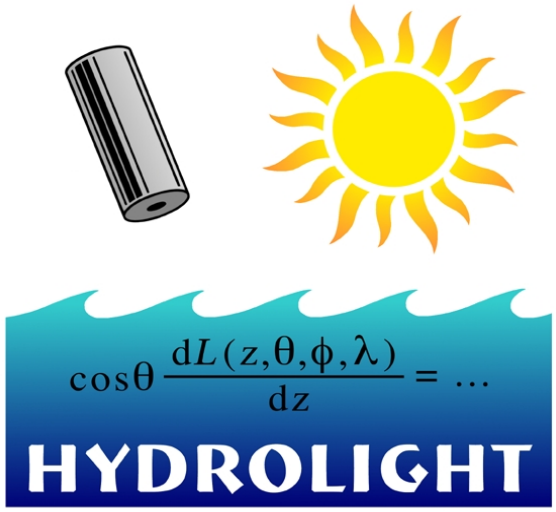

HYDROLIGHT 5.2
ECOLIGHT 5.2

TECHNICAL DOCUMENTATION

Curtis D. Mobley
Lydia K. Sundman
Sequoia Scientific, Inc.

SEQUOIA

First Printing, October 2013

Update Note

This version of the the HydroLight-EcoLight Technical Documentation updates the previous version of August 2012 (for HydroLight-EcoLight version 5.1) to cover changes up to and including code versions 5.2.0. A pdf of this document with color figures is in the HE52/Documents directory.

Copyright

HydroLight and EcoLight versions 5.2 are copyright ©2013 by Curtis D. Mobley. All rights reserved worldwide. This Technical Documentation is protected by United States copyright law. No part of this document may be copied or distributed, transmitted, transcribed, stored in a retrieval system, or translated into any human or computer language, in any form or by any means (electronic, mechanical, magnetic, manual, or otherwise), except by prior written permission of Curtis D. Mobley. Licensees of HydroLight and EcoLight are given limited permission to reproduce this document provided such copies are for their use only and are not sold or distributed to third parties. All such copies must contain the title page and this notice page in their entirety.

Trademarks

Microsoft, MS-DOS, Visual Basic, Excel, and Windows are registered trademarks of Microsoft Corporation. Lahey is a registered trademark of Lahey Computer Systems, Inc. IDL is a registered trademark of Exelis Visual Information Solutions, Inc. All other brand names are trademarks of their respective holders.

Disclaimer

Curtis D. Mobley and Sequoia Scientific, Inc. reserve the right to revise the HydroLight and EcoLight software, Users' Guide, Technical Documentation, or other publications and products with no obligation to notify any person or any organization of such revision. In no case shall Curtis D. Mobley or Sequoia Scientific, Inc. be liable for any loss of profit or any other commercial damage, including but not limited to special, consequential, or other damages.

Technical Support

Technical support for HydroLight and EcoLight can be obtained in accordance with the user's license agreement from

Curtis D. Mobley
425-641-0944 ext 109
fax: 425-643-0595
curtis.mobley@sequoiasci.com

Lydia K. Sundman
lsundman@sequoiasci.com

If you encounter a problem during a HydroLight or EcoLight version 5 run, please e-mail us the following:

- The "Licensed to" name and Serial Number of the copy you are running. These are found on the distribution CD, on opening form of the User Interface, and at the top of the printout.
- A description of the problem, error messages, or other pertinent information.
- The input file (the `Iroot.txt` file for the run, which is found in the HE52\run\batch directory)
- The printout file (the `Proot.txt` file, from either the HE52\output\Hydrolight\printout or HE52\output\Ecolight\printout directories)
- Any user-supplied input data files used by the run.
- If your run included a user-defined IOP model or function, or if you included bioluminescence, please also send us `root.for` file from the HE52\code\batch directory and any user-defined subroutines we would need to repeat the run.

This information will greatly increase the speed at which we can troubleshoot the problem.

HydroLight Users' Group

HydroLight licensees are strongly encouraged to join the HydroLight Users' Group at http://tech.groups.yahoo.com/group/HydroLightUsers/. This web site is used to make announcements about HydroLight, distribute detailed technical notes about HydroLight algorithms, post bug fixes, share new code written by users, and for communication of general interest between HydroLight users. A file of FAQs is maintained there.

Acknowledgment

Dr. Marcos Montes of the U.S. Naval Research Laboratory provided invaluable assistance in extending the RADTRAN sky irradiance model to 300 and 1000 nm. He has also assisted with beta testing and developed several internal changes to the core code, which have improved its numerical efficiency and accuracy. Dr. Eric Rehm also performed very useful beta testing of the EcoLight code and comments on Appendix A.

TABLE OF CONTENTS

1. INTRODUCTION. 1
   1.1 HydroLight Technical Notes. 2

2. IOP MODELS. 3
   2.1 Pure Water. 6
   2.2 Constant. 7
   2.3 "Classic" Case1. 7
   2.4 "New" Case1. 10
      2.4.1 Particle Absorption. 10
      2.4.2 CDOM Absorption. 14
      2.4.3 Scattering. 16
      2.4.4 Scattering Phase Function. 18
   2.5 Case2. 21
   2.6 Measured IOP Data. 21
   2.7 User-written IOP Models. 25
   2.8 Specific-Absorption Models. 25
   2.9 Specific-Scattering Models. 28

3. BOTTOM-REFLECTANCE MODELS. 30
   3.1 Finite-depth Water. 30
   3.2 Infinitely Deep Water Without Inelastic Scattering or Internal Sources. 31
   3.3 Infinitely Deep Water with Inelastic Scattering or Internal Sources-- Dynamic Bottom Boundary Depths. 33

4. SEA-SURFACE MODEL. 35

5. SKY MODELS. 37
   5.1 The RADTRAN-X Sky Irradiance Model. 38
   5.2 The Empirical Sky Radiance Model. 42
   5.3 The Analytic Sky Radiance Model. 44
   5.4 Input of User-defined Sky Irradiances. 44
      5.4.1 Running HE52 outside the 300-1000 nm range. 45
      5.4.2 Simulation of Lidar-induced inelastic scatter. 49

6. INELASTIC-SCATTERING MODELS. 51

7. STANDARD-FORMAT DATA FILES. 52
   7.1 Concentration Data. 53
   7.2 Concentration-specific Absorption and Scattering Data. 54
   7.3 Absorption and Beam-attenuation Data. 55

7.4 Backscatter Data. . . . 57
7.5 Bottom Reflectance Data. . . . 57
7.6 Sky Irradiance Data. . . . 58
7.7 Biolumnescence Data. . . . 60
7.8 Filling in Missing Data. . . . 61
7.8.1 Interpolation within the measurement range. . . . 61
7.8.2 Extrapolation outside the measurement range. . . . 62

8. WRITING SUBROUTINES FOR IOP AND OTHER MODELS. . . . 64
8.1 Subroutines for IOP Models. . . . 65
8.2 Subroutines for Chlorophyll Profiles. . . . 68
8.3 Subroutines for CDOM Absorption. . . . 69
8.4 Subroutines for Bottom BRDFs. . . . 69
8.5 Subroutines for Sky Models. . . . 70
8.6 Subroutines for Bioluminescence. . . . 70

9. SPECIAL RUNS. . . . 71
9.1 Defining New Quad and Band Partitions and Creating New Surface Files. . . . 72
9.2 Discretizing a Phase Function. . . . 73
9.3 High-theta-resolution EcoLight runs. . . . 78

10. WHEN SOMETHING GOES WRONG. . . . 81

11. REFERENCES. . . . 85

APPENDIX A. DETAILED DESCRIPTION OF THE RUN-TIME INPUT FOR STANDARD RUNS. . . . 87

APPENDIX B. DESCRIPTION OF ARRAYS AND I/O UNIT NUMBERS. . . . 103
B.1 The IMISC Array. . . . 103
B.2 The FMISC Array. . . . 104
B.3 The DATAFILES Array. . . . 105
B.4 Reserved I/OUnit Numbers. . . . 106

APPENDIX C. LICENSE AGREEMENT FOR USE OF HYDROLIGHT-ECOLIGHT VERSION 5. . . . 108

1. INTRODUCTION

The separate Users' Guide gives a general overview of the capabilities of HydroLight-EcoLight version 5.2, describes how to install and run the code, and shows example output. This Technical Documentation is designed to provide more detailed information for users who are already familiar with the basic operation of HydroLight-EcoLight version 5.2. For brevity, HydroLight-EcoLight version 5.2 will be called HE52 when referring to common features of the codes. Differences in HydroLight and EcoLight will be noted as necessary, in which case the individual names will be used. This documentation includes descriptions of various models available within HE52 and instructions on how to customize HE52 for a user's specific purpose. Formats for data files are described, as are procedures for creating new phase function and sea-surface files.

This document assumes that the reader has basic familiarity with HE52. If new to HE52, at least run the examples found in the Users' Guide to get an idea of what input is required and how it can be specified. It is also assumed that the reader is familiar with the basic terminology and notation of optical oceanography. If this is not the case, then the reader should first consult the review paper by Mobley (1995) or one of the books by Kirk (1994); Spinrad, Carder, and Perry (1994); or Mobley (1994). The "Ocean Optics Web Book" at http://www.oceanopticsbook.info/ also contains much useful infomation. The text *Light and Water: Radiative Transfer in Natural Waters* (Mobley, 1994) describes in considerable detail the mathematical methods employed in HydroLight. That book is the primary reference for HydroLight. The source code itself is documented by references to the equations of *Light and Water*.

Throughout this report, the names of mathematical variables are written in italics, e.g., *U*, *z*, or *zeta*. The names of computer programs, directories, and files are written in a sans serif font, e.g., abcase1.f or Pupcast2.txt. Path names are written using the Microsoft Windows DOS format with a backstroke, e.g., ..\data\phasefun\hydrolight\avgpart.dpf, which is appropriate for PC operating systems. User input to and output from programs is shown in `Courier`. Options on the Graphical User Interface are shown in SMALL CAPS.

Most of the figures in this Technical Documentation were generated in color but printed in black and white for the hard copy. The color figures can be seen on the pdf version of the Technical Documentation, which is found in the HE52\documents directory on the distribution CD.

1.1 HydroLight Technical Notes

HydroLight and EcoLight users often pose excellent questions about how various computations are done, how to interpret outputs, and the like. Questions of general interest requiring detailed explanations sometimes result in a HydroLight Technical Note (HTN). These HTNs are found in the HE52\Documents directory (and on the HydroLight Users' Group website) and supplement the information in this Technical Documentation. The HTNs issued to date are

Note	Date	Topic
—	Nov 2004	How Secchi depth is computed
—	Apr 2005	Informal notes on various types of reflectance, including how BRDFs are used in HydroLight and Monte Carlo codes
HTN1	Apr 2005	How H simulates wind-blown sea surfaces
HTN2	Sept 2002	How H output depends on angular resolution
HTN3	Sept 2002	Advice on choosing output depths.
HTN4	Dec 2002	The RADTRAN-X atmospheric model
HTN5	Jan 2007	How H spectral outputs are converted to visual color as CIE chromaticity coordinates and RGB values
HTN6	May 2008	Advice on selecting bottom depth in runs with inelastic scattering. Note: this advice has now been automated in the code via the dynamic lower boundary depth option, so this note is now obsolete.
HTN7	Aug 2008	Conservation of energy across the sea surface in H
HTN8	Oct 2010	How H computes Forel-Ule color indices
HTN9	Dec 2010	New options for input of user-defined sky irradiances begun in version 5.1.2. Note: these options are now discussed in Section 5.4 of this Tech Doc, so this HTN is now obsolete.
HTN10	Jan 2012	Interpretation of Raman scatter output

Table 1. HydroLight Technical Notes issued through October 2013.

2. IOP MODELS

HE52 comes with several built-in models for describing water inherent optical properties (IOPs), namely the water absorption coefficient *a* and scattering coefficient *b* as functions of depth and wavelength, and the scattering phase function, which depends on scattering angle and possibly also on depth and wavelength. These models are accessible from within the graphical user interface (UI) and can be used as is. These models also can be the starting point for creating models tailored to a particular user's specific situation. This section briefly describes the IOP models available in the HE52 software package.

All IOP routines have the same fundamental structure (see HE52\examples\templates\ab.f), which allows the same subroutine to be used by both HydroLight and EcoLight. The source codes for these and other subroutines that are common to both HydroLight and EcoLight are in the HE52\code\common directory. The UI prompts the user for the various inputs needed by these routines. The IOP routines allow users to model the IOPs of almost any water body. For example, the user can

- select models for Case 1 water, which are based on the user-supplied chlorophyll concentration. These models convert the user-supplied, depth-dependent chlorophyll concentration to depth- and wavelength-dependent absorption (*a*) and scattering (*b*) coefficients.

- build up the *a* and *b* values for Case 2 water by defining the concentrations and specific absorption and scattering coefficients of non-covarying microbial, dissolved, and mineral components.

- obtain depth- and wavelength-dependent *a* and *b* values from user-supplied data files containing measured *a* and *c* values (e.g, data from a WETLabs ac-9 or similar instrument; *c* is the beam attenuation coefficient and $b = c - a$). Depth- and wavelength-dependent scattering phase functions can be determined from associated measurements of the backscatter coefficient, if available.

- provide input to user-written IOP models with any number of components.

The total IOPs of a water body are built up as a sum of IOPs attributable to the various components of the water body. Thus, the total absorption coefficient is computed from (*Light and Water*, Eq. 3.10)

$$a_{\text{total}}(z,\lambda) = \sum_{i=1}^{ncomp} a_i(z,\lambda) .$$

Here $a_i(z,\lambda)$ is the absorption coefficient of the i^{th} component of the water body (e.g., pure water, chlorophyll-bearing phytoplankton, CDOM, or mineral particles), which in general is a function of the depth z and wavelength λ. The number of components in the IOP model is *ncomp*. A similar equation is used to compute the total scattering coefficient b. To complete the specification of the water IOPs, a scattering phase function must be specified for each component. In some IOP models, the user explicitly selects the phase function to be used for each component; in others the phase functions are pre-determined. The Measured IOP data model also has the option of allowing the phase function to be determined dynamically (i.e., as HydroLight or EcoLight runs to solve the radiative transfer equation) from depth- and wavelength-dependent scattering and backscattering coefficients read from user-supplied data files.

Table 1 lists the various IOP models available in HE52. These models are then described.

name	components	comments
pure	1: pure water	returns *a* and *b* values for pure water (called by other IOP models)
constant	1: the total *a* and *b* values	returns depth-independent *a* and *b* for a single wavelength
"Classic" Case 1	1: pure water 2: particles 3: co-varying CDOM	"Gordon-Morel" Case 1 water model; requires the chlorophyll concentration and particle phase function as input. This is the Case 1 model found in HydroLight version 4.
"New" Case 1	1: pure water 2: small chlorophyll-bearing particles 3: large chlorophyll-bearing particles 4: co-varying CDOM	a NEW Case 1 water model based on recent bio-optical models for absorption and scattering; requires only the chlorophyll concentration as input. Recommended as a replacement for the classic Case 1 model of version 4.
Case 2	1: pure water 2: chlorophyll-bearing particles 3: CDOM 4: mineral particles	a generic 4-component IOP model with great flexibility in how the user can define component optical properties. Recommended for general use.
Measured IOP data	1: pure water 2: everything else, (e.g., as measured by an unfiltered ac-9)	Reads standard-format (see §7.3) data files to obtain *a* and *b*. Optionally, an additional data file can be read to obtain the backscatter coefficient b_b for use in determining the phase function.
User-written	any number of components (default 4; can be increased in the User Interface)	Allows the user to access user-written IOP subroutines from within the HE52 UI.

Table 1. The IOP models for absorption *a* and scattering *b* provided with HE52. The source code is in the HE52\code\common directory with filename prefixed with "ab", e.g., the new Case 1 model is in routine abnewCase1 found in file HE52\code\common\abnewCase1.f.

2.1 PURE WATER. This IOP routine returns the absorption coefficient *a* and scattering coefficient *b* for "pure" water, as selected by the user in the UI. The options available in the UI have *a* values based on Pope and Fry (1997) and scattering for either pure fresh water or pure sea water, or have absorption from Smith and Baker (1981) and scattering by sea water. (The Pope and Fry and Smith and Baker absorptions are both appended by other data sets as needed for UV and IR wavelengths.). A user-created data file of *a* and *b* values also can be read. The Pope and Fry data are read from file HE52\data\H20abDefaults_SEAwater.txt or HE52\data\H20abDefaults_FRESHwater.txt ; the Smith and Baker data are read from file HE52\data\H20abClearNat.txt. The header records in these two data files give the references for the data sets used to extend the IOPs beyond the wavelengths of the original data sets. Figure 1 shows these spectra. The scattering coefficients for pure sea water are computed from (Morel, 1974; c.f. *L&W* page 103):

$$b = 16.06 \cdot 1.80 \cdot 10^{-4} \cdot (500.0/\lambda)^{4.32} ,$$

with a similar formula for fresh water. The scattering phase function $\tilde{\beta}(\psi)$ for both pure sea water or pure fresh water is (*L&W* Eq. 3.30) $\tilde{\beta}(\psi) = 0.06225(1 + \cos^2\psi)$.

.

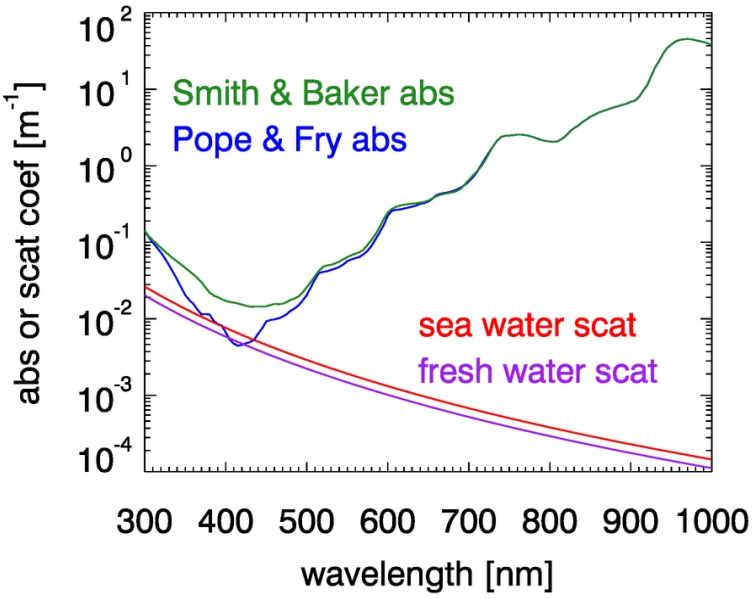

Fig. 1. The pure water IOPs available for selection in the HE52 UI.

The pure water IOP model is not an explicit option on the UI because it is of no interest to most users. However, simulations for pure water can be obtained simply by using the classic Case 1 IOP model and using a chlorophyll concentration of 0. The same pure water IOPs are accessed within each IOP model by calls to subroutine HE52\code\common\abpure.f, which reads the chosen pure water data file and can be used for the same purpose by user-written IOP subroutines

2.2 CONSTANT. This IOP model simply gives the *total* (including water) *a* and *b* values at a single wavelength for a homogeneous water body. It is designed for idealized radiative transfer studies and pedagogical purposes rather than for modeling actual water bodies. This is the only IOP model that allows the depth to be measured as non-dimensional optical depth ζ, rather than as geometric depth z in meters. The UI requires the user to pick the total scattering phase function. Because constant is for runs at a single (user-selected) wavelength, various options (such as the inclusion of inelastic scatter) are automatically disabled if this IOP model is used.

2.3 "CLASSIC" CASE 1. This model is based on a recent reformulation (Morel and Maritorena, 2001) of the Case 1 water model seen in *Light and Water*, Eqs. 3.27 and 3.40. The absorption coefficient is modeled as the sum of three components:

$$a_{total}(z,\lambda) = a_w(\lambda) + a_p(z,\lambda) + a_y(z,\lambda), \quad (1)$$

where a_w is absorption by pure water, a_p is absorption by chlorophyll-bearing particles, and a_y is absorption by covarying yellow matter (CDOM). The particle absorption is given by

$$a_p(z,\lambda) = 0.06\, A_{chl}(\lambda)\, [Chl(z)]^{0.65}, \quad (2)$$

where $Chl(z)$ is the user-supplied chlorophyll concentration profile in mg Chl m^{-3}, and $A_{chl}(\lambda)$ is the non-dimensional chlorophyll-specific absorption coefficient given in Prieur and Sathyendranath (1981; their $a_c^{*\prime}$) and Morel (1988, his Fig 10c) as extrapolated to 300 and 1000 nm. This $A_{chl}(\lambda)$ is shown in Fig. 2. It should be noted that this $A_{chl}(\lambda)$ is independent of the chlorophyll concentration. Thus only the magnitude of the particle absorption coefficient depends on *Chl*; the shape of the a_p spectrum is the same for all *Chl* values. Our

extrapolation of the original $A_{chl}(\lambda)$ to 300 and 1000 nm is somewhat uncertain (especially at UV wavelengths; see §2.4.1 below). We therefore recommend the "new Case 1" IOP model for general use, especially near 300 nm. The "classic" Case 1 IOP model is retained in HE52 for ease of comparison with the "new" Case 1 IOP model.

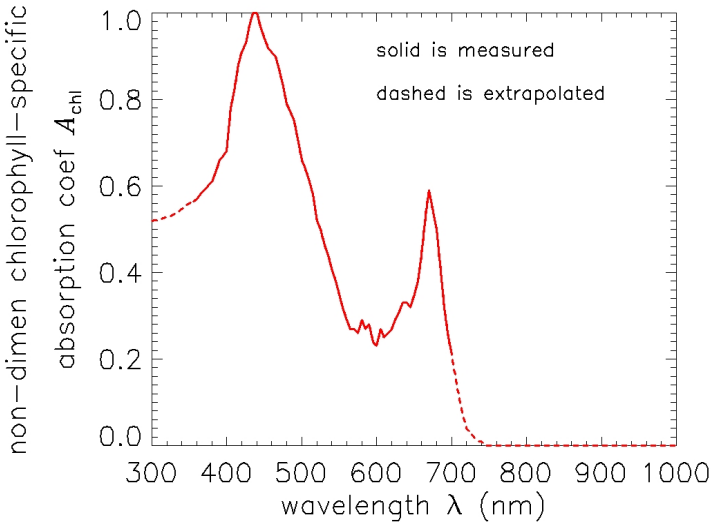

Fig. 2. The non-dimensional chlorophyll-specific absorption coefficient $A_{chl}(\lambda)$ used in Eq. (2).

Absorption by yellow matter (colored dissolved organic matter, or CDOM) covaries with particle absorption according to

$$a_y(z,\lambda) = 0.2\, a_p(z, 440\ \text{nm}) \exp[-0.014\,(\lambda - 440)]. \tag{3}$$

The scattering coefficient for the particles is given by

$$b_p(z,\lambda) = b_o [Chl(z)]^n \left(\frac{550}{\lambda}\right)^m, \tag{4}$$

where the defaults are $b_o = 0.3$, $n = 0.62$, and $m = 1$ (Gordon and Morel, 1983, or *L&W* Eq. 3.40). The yellow matter is assumed to be non-scattering.

The user must specify both the chlorophyll concentration $Chl(z)$ and the scattering phase function for the particles. One way to specify the particle phase function is to give the

particle backscatter fraction, $B_p = b_{bp}/b_p$, from which HE52 generates a Fournier-Forand phase function with the requested value of B_p (Mobley, et al., 2002). Various formulas for B_p as a function of chlorophyll can be found in the literature. For example, Ulloa et al. (1994) give an empirical formula for B_p at 550 nm in Case 1 waters,

$$B_p = 0.01\,[0.78 - 0.42 \log_{10}(Chl)], \tag{5}$$

and Twardowski et al. (2001) present another formula,

$$B_p = 0.0096\, Chl^{-0.253}.$$

Such formulas for B_p can be used for rough guidance in specifying the phase function, in the absence of other information about the phase function. However, it must be remembered that B_p correlates poorly with *Chl*, and there can be order-of-magnitude variability in the measured value of B_p for a given *Chl* value in a particular data set.

The particle backscatter fraction can chosen to be the same at every wavelength, or be a function of wavelength according to the power law

$$B_p(\lambda) = B_p(\lambda_o)\left(\frac{\lambda_o}{\lambda}\right)^n.$$

Values for λ_o, $B_p(\lambda_o)$, and n are entered in the User Interface. If it is desired to have the particle backscatter fraction be a function of both depth and wavelength, then the IOP data model described in §2.6 must be used to read in separate files of $b_p(z,\lambda)$ and $b_{bp}(z,\lambda)$ such that $B_p(z,\lambda) = b_{bp}(z,\lambda)/b_p(z,\lambda)$ has the desired values.

Note that the absorption model of Eq. (1) reduces to that of pure water if *Chl* = 0, which is slightly different than some formulations (e.g., Morel and Maritorena, 2001, Eq. 16-18; *L&W*, Eq. 3.27), which include a small amount of background CDOM absorption even in the absence of phytoplankton. (The inclusion of a small amount of background CDOM is reasonable, but in all honesty, we just got tired of explaining to HydroLight users why they didn't get exactly the same results as for pure water when they plugged in *Chl* = 0 in the Case 1 IOP model. In any case, the difference is negligible except at extremely low *Chl* values, in which case the model is suspect anyway.)

This IOP model is retained in HE52 for comparison purposes. It is called the "Classic Case 1" IOP model in the HE52 user interface.

2.4 "NEW" CASE 1. This is a new IOP model, which is based on recent publications on absorption and scattering in Case 1 waters. We therefore describe it in some detail for comparison with the well-known "classic" IOP model just described, which has been used by researchers for decades.

2.4.1 Particle absorption. It is well known that there is great variability in chlorophyll-specific absorption spectra $a^*(\lambda)$. In particular, the spectral shape of $a^*(\lambda)$ changes with the chlorophyll concentration, owing to species composition and pigment packaging effects (e.g., Bricaud et al., 1995, 1998). Thus the next step in improving the particle absorption model is to allow the chlorophyll-specific absorption $a^*(\lambda)$ to depend on the chlorophyll concentration itself. Bricaud et al. (1998) therefore model particle absorption as

$$\begin{aligned} a_p(z,\lambda) &= a^*(Chl,\lambda)\, Chl(z) \\ &= A(\lambda)\, [Chl(z)]^{-B(\lambda)}\, Chl(z) \\ &= A(\lambda)\, [Chl(z)]^{E(\lambda)} . \end{aligned} \tag{6}$$

The Bricaud et al. (1998) paper gives $A(\lambda)$ and $E(\lambda)$ between 400 and 700 nm. Extending the Bricaud et al. values from 700 to 1000 nm is easy because phytoplankton absorption is essentially zero in the IR. However, there are very few measurements of phytoplankton absorption below 350 nm, so extending $A(\lambda)$ and $E(\lambda)$ down to 300 nm is an uncertain process.

Morrison and Nelson (2004) [their Fig. 1] show two normalized phytoplankton absorption spectra from 300 to 750 nm taken at the Bermuda Atlantic Time Series (BATS) site. The BATS *Chl* values ranged between 0.002 and 0.606 over the course of a year, with a mean of 0.152 mg Chl m^{-3}. Although their spectra are similar above 365 nm, they are highly variable with season and depth between 300 and 365 nm. This variability is likely due to mycosporine-like amino acids (MAAs), which strongly absorb near 320 nm. Figure 3 compares the Morrison and Nelson spectra (blue curves) with the Bricaud et al. a_p of Eq. (6) evaluated for $Chl = 0.05$ mg m^{-3} (red curve); the Morrison and Nelson spectra are normalized to the Bricaud value of $a_p(400)$. The shapes of the Morrison and Nelson spectra are consistent with the Bricaud values for low *Chl* values.

Vasilkov et al. (2005) present spectra for $A(\lambda)$ and $B(\lambda) = 1 - E(\lambda)$ between 300 and 400 nm, as derived from absorption measurements in coastal California waters. Figure 4 shows their $A(\lambda)$ and $B(\lambda)$ spectra compared with those of Bricaud et al. (1998). The differences at 400 nm reflect the different databases (i.e., different waters) used to derive the coefficients.

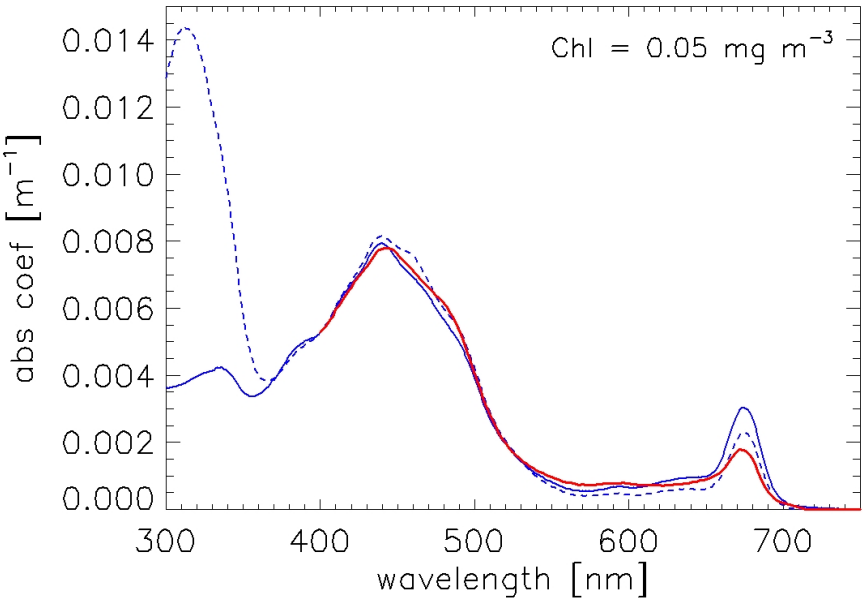

Fig. 3. Comparison of Bricaud et al. a_p for *Chl* = 0.05 (red) with the Morrison and Nelson normalized absorption spectra (blue; dotted is summer, solid is winter).

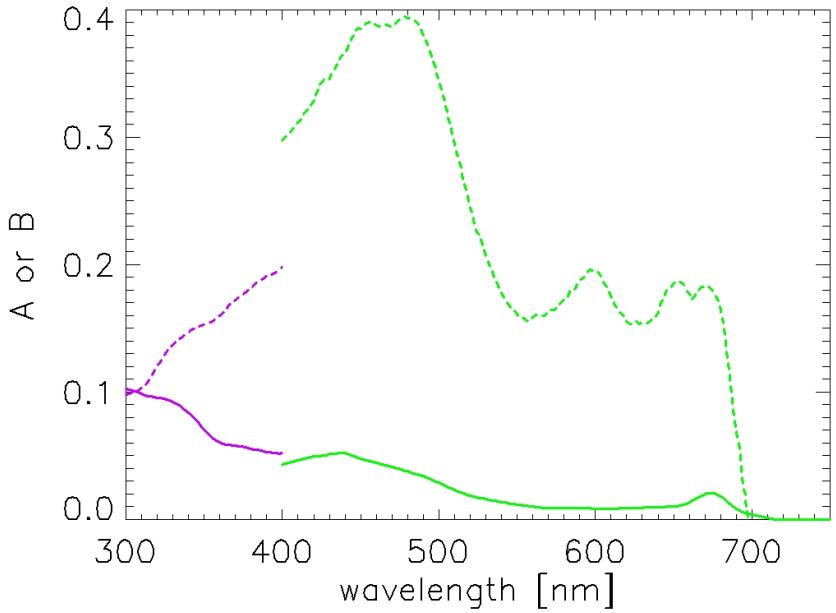

Fig. 4. Comparison of the Vasilkov et al. (2005) $A(\lambda)$ and $B(\lambda)$ (purple curves) with those of Bricaud et al. (1998) (green curves). Solid lines are *A* and dashed lines are *B*.

Desperation is the mother of modeling (you can quote us on that), so to define $A(\lambda)$ and $B(\lambda)$ spectra over the 300-1000 nm range, we proceeded as follows. The Bricaud et al. A and B curves are accepted for use from 400 to 720 nm, with $A = B = 0$ between 720 and 1000 nm. The Vasilkov et al. curve for A was normalized to the Bricaud value at 400 nm, i.e., $A_v(\lambda) \leftarrow [A_v(\lambda)/A_v(400)]A_b(400)$, where subscripts v and b stand for Vasilkov et al. and Bricaud et al., respectively. The normalized $A_v(\lambda)$ was then averaged with the two normalized spectra of Morrison and Nelson seen in Fig. 1, *assuming* that the A spectra have the same shape as a_p. This assumption about the shapes of A and a_p is correct only if $B = 0$ or if $Chl = 1$, in which case $A = a^*$ in Eq. (6). The resulting average A between 300 and 400 nm then merges smoothly with the A of Bricaud at 400 nm. For B, the Vasilkov et al. curve was normalized to the Bricaud et al. curve at 400, and the result was used to extend the Bricaud et al. B down to 300 nm. The resulting A and B spectra are shown in red in Fig. 5, along with the three A spectra used to compute the average A between 300 and 400. These A and B give an absorption model that roughly corresponds to the mid-range of UV absorptions seen in the Morrison and Nelson data. The new Case 1 IOP model uses these A and $E = 1 - B$ as the default spectra for the last version of Eq. (6). (The tabulated A and E spectra are on file data\AE_midUVabs.txt.)

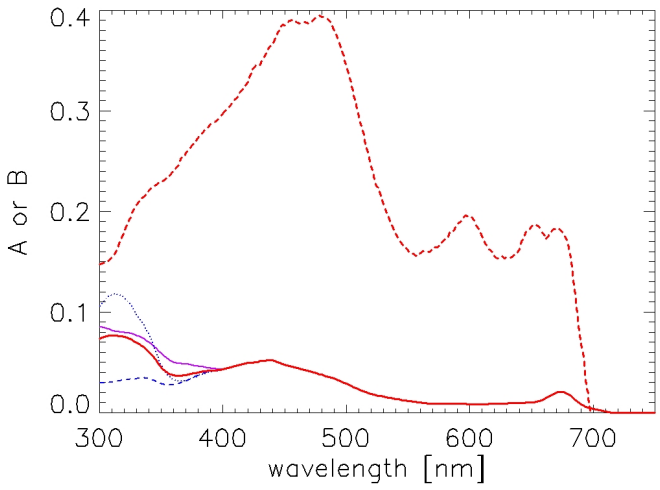

Fig. 5. The A (red solid line) and B (dashed line) spectra used in Eq. (6) to define a_p for the mid-range of UV absorption. The purple and blue curves were averaged to produce A between 300 and 400 nm.

Some HE52 users will likely be interested in comparisons of underwater light fields for the wide range of UV absorptions illustrated by the Morrison and Nelson spectra of Fig. 3. We therefore defined *A* spectra for low and high UV absorptions by simply using the shapes of the Morrison and Nelson spectra absorption spectra to extend the Bricaud *A* from 400 down to 300 nm. The *B* spectra were taken to be the same as for the mid-range of UV absorption just discussed. The user can select low- (file AE_lowUVabs.txt) , mid-, or high-UV (file AE_highUVabs.txt) absorptions in the HE52 UI. Users who have their own data for *A* and *B* can place their data in an ASCII file of the same format as AE_midUVabs.txt and select the new file in the UI. *If the user has a measured a^* spectrum and wishes to use it in Eq. (6), the a^* can be entered in the file as A, and E can be set to 1.* The Bricaud model of Eq. (6) then reduces to the classic IOP phytoplankton absorption model with the form

$$a_p(z,\lambda) = a^*(\lambda)\, Chl(z)\,.$$

Regardless of which set of *A* and *E* spectra is chosen, the *A* and *E* spectra are used in the same manner in Eq. (6) to define $a_p(\lambda)$ for any *Chl* value. Figure 6 shows the resulting particle absorption spectra for low, medium, and high UV absorptions and for *Chl* = 0.01, 0.1, 1.0, and 10.0 mg m^{-3}. The corresponding absorption coefficients as computed by the classic Case 1 IOP model are shown for comparison, as is absorption by pure water. There are significant differences in the classic and new models, which will lead to significant differences in computed radiances, irradiances, and AOPs when used in HE52. Note in particular that the shape of the particle absorption spectrum now changes with the chlorophyll value. Presumably the new model gives a more realistic description, on average, of particle absorption in Case 1 waters than does the classic model of Eq. (2).

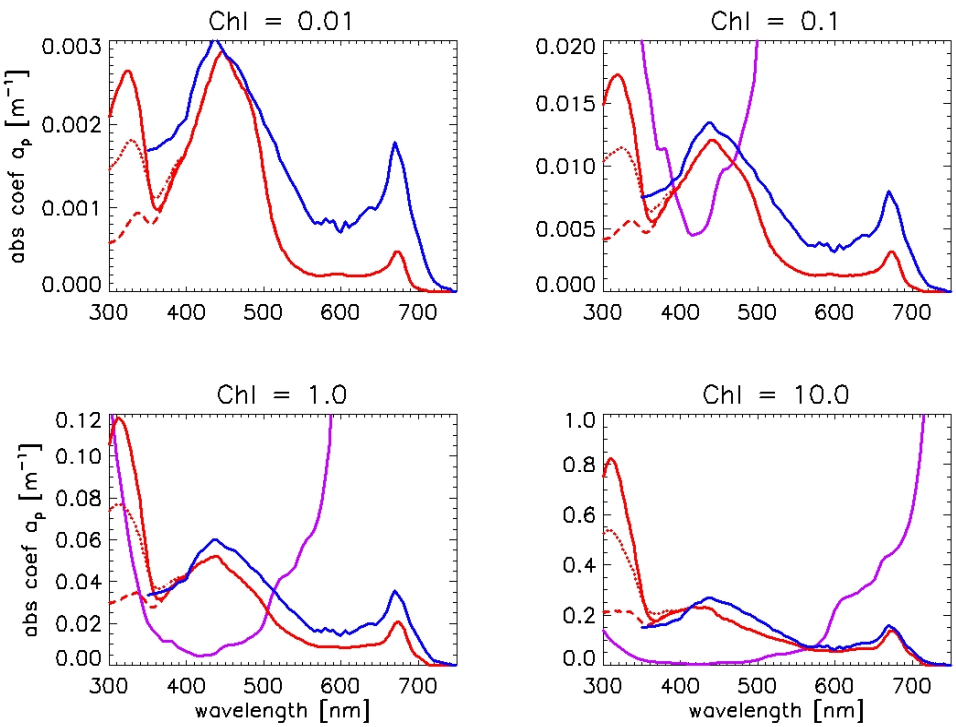

Fig. 6. Particle absorption coefficients computed by the "new Case 1" model of Eq. (6) (red) and the "classic Case 1" model of Eq. (2) (blue). The red solid line is for the high-UV absorption, dotted is the mid-range UV absorption, and dashed is the low-range UV absorption. The purple line is absorption by pure water.

2.4.2 CDOM absorption. Figure 7 shows measured CDOM absorption spectra down to 200 nm at several locations in Florida waters on both log and linear ordinates (data of Lore Ayoub, personal communication). For wavelengths greater than 300 nm, these spectra are acceptably well modeled by a function of the form of Eq. (3):

$$a_y(z,\lambda) = a_y(z,\lambda_o)\exp[-S(\lambda-\lambda_o)]. \quad (7)$$

The red line in Fig. 7 shows the spectrum predicted by Eq. (7) with $\lambda_o = 440$ nm, $S = 0.0162$ nm^{-1} and the average (for the spectra shown) of $a_y(440)$. This value of S was determined from the average values of a_y at 300 and 440 nm. The functional form (7) is used to model CDOM absorption down to 300 nm, as needed for HE52. This model underestimates CDOM absorption at shorter wavelengths, but that is irrelevant for HE52. When incorporated into the new Case 1 IOP model, $a_y(z,\lambda_o)$ is set to $f_y\, a_p(z,\lambda_o)$, with default values

of $f_y = 0.2$, $\lambda_o = 440$ nm, and $S = 0.014$ nm^{-1} just as in Eq. (3) for the classic Case 1 model. However, the user can change the values of f_y, λ_o, and S in the HE52 UI if other values are desired.

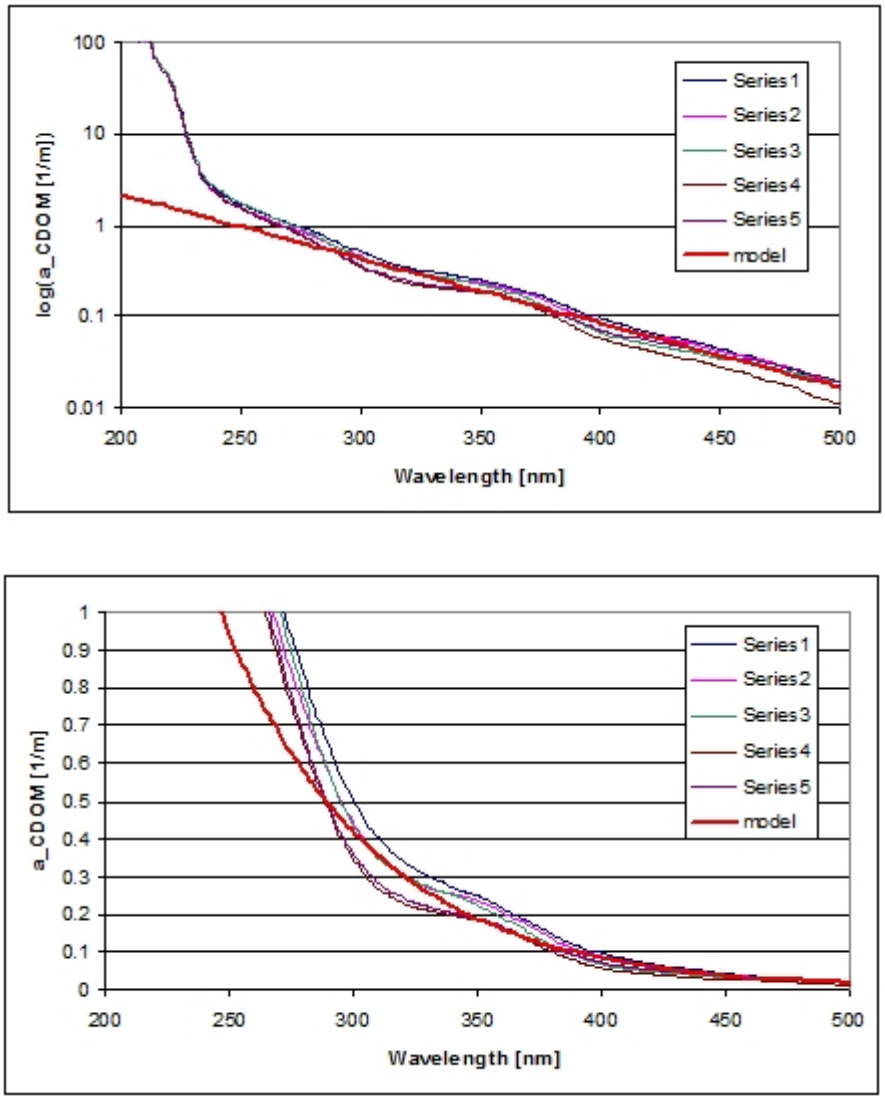

Fig. 7. Measured and modeled (red) CDOM absorptions.

2.4.3. Scattering. Just as for absorption, recent papers have presented improved models for particle scattering in Case 1 waters. Morel et al. (2002, Eq. 14) model the particle scattering coefficient as

$$b_p(z,\lambda) = b_o[Chl(z)]^n \left(\frac{\lambda}{550} \right)^{\nu}, \tag{8}$$

where

$$\nu = 0.5\,[\log_{10} Chl - 0.3] \quad \text{for } 0.02 < \text{Chl} < 2$$
$$= 0 \text{ for Chl} > 2\,.$$

Thus the wavelength dependence of the scattering coefficient now depends on the chlorophyll concentration, unlike in the classic model of Eq. (4). In particular, υ now lies between -1 and 0. A value of $\upsilon = -1$, as seen in Eq. (4), is known from Mie theory to be valid for nonabsorbing particles with a Junge particle size distribution of slope -4.

Loisel and Morel (1998) studied the relationship between particle beam attenuation at 660 nm, $c_p(660)$, and the chlorophyll concentration. They found the functional form

$$c_p(z,660) = c_o[Chl(z)]^n\,. \tag{9}$$

The values of c_o and n are different for near-surface (down to one "penetration depth," as relevant to remote sensing) and deeper waters. Because $c_p \approx b_p$ at 660, Morel et al. (2002) adopt the coefficients for Eq. (9) for use in Eq. (8), after shifting the reference wavelength to 550 nm. Thus for near surface waters, Morel et al. (2002) use $b_o = 0.416$ and $n = 0.766$ in Eq. (8).

However, a power law in wavelength of the form of Eq. (8) generally gives a better fit for c_p than for b_p (e.g., Voss, 1992; Boss et al., 2001). Thus it is probably better to model c_p and then obtain b_p from $b_p = c_p - a_p$ (with a_p being determined by Eq. (6) as described above). This is the approach taken in the new Case 1 model, which uses

$$c_p(z,\lambda) = c_o[Chl(z)]^n \left(\frac{\lambda}{660} \right)^{\nu}, \tag{10}$$

where

$$\nu = 0.5\,[\log_{10} Chl - 0.3] \quad \text{for } 0.02 < \text{Chl} < 2$$
$$= 0 \text{ for Chl} > 2\,.$$

Thus the new model uses the chlorophyll-dependence of $c_p(660)$ from Loisel and Morel (1998), Eq. (9), and assumes that c_p has the same chlorophyll-dependent wavelength dependence as the b_p model of Morel et al. (2002), Eq. (8). The values of c_o and n can be chosen by the user in the HE52 UI. The default values, which apply to near-surface waters, are $c_o = 0.407$ and $n = 0.795$ (from Loisel and Morel, 1998, Eq. 5).

Figure 8 shows example a_p for mid-range UV absorption, b_p, and c_p spectra for near-surface waters ($c_o = 0.416$ and $n = 0.766$ in Eq. 10), along with the "classic" b_p of Eq. (4) with $b_o = 0.3$, $n = 0.62$, and $m = 1$. The scattering coefficients are not too different at low chlorophyll values, but the new b_p has a different wavelength dependence and is much larger in magnitude, by up to a factor of three, at high *Chl* values. Unlike in the classic scattering model of Eq. (4), the wavelength dependence of b_p now depends on *Chl* and is more complicated. These differences in scattering will have a significant effect on computed radiances.

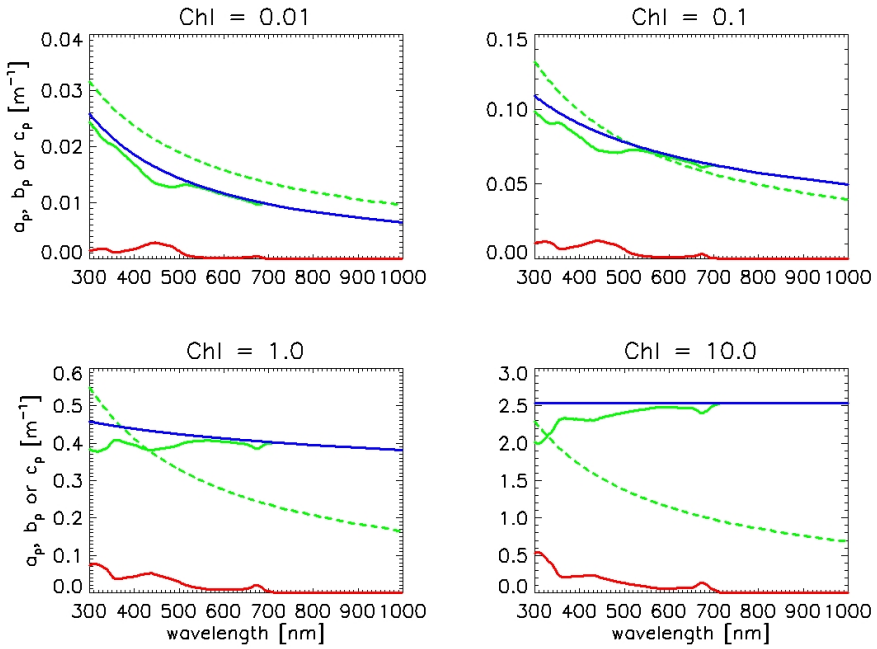

Fig. 8. Beam attenuation c_p as determined by from Eq. (10) and near-surface values of $c_o = 0.407$ and $n = 0.795$ (blue). The red curves are the same a_p as the mid-UV absorptions in Fig. 6. The solid green curve is the new $b_p = c_p - a_p$; the dashed green curve is the "classic" b_p of Eq. (4).

2.4.4. Scattering phase function. When using the classic Case 1 IOP model, the user must also specify the scattering phase function to be used for particle scattering. This can be done in various ways, e.g. by picking a particular phase function (such as the Petzold "average-particle" phase function defined in Mobley, et al., 1993), or by specifying the particle backscatter fraction $B_p = b_{bp}/b_p$, which is then used to generate a Fourier-Forand phase function with that backscatter fraction (Mobley, et al., 2002).

Morel et al. (2002) developed a phase function model for Case 1 water in which the phase function is a combination of "small" and "large" particle phase functions, with the fraction of each being determined by the chlorophyll concentration:

$$\tilde{\beta}_p(\psi,Chl) = \alpha_s(Chl)\,\tilde{\beta}_s(\psi) + \alpha_l(Chl)\,\tilde{\beta}_l(\psi)\,, \tag{11}$$

where

$$\alpha_s(Chl) = 0.855\,[0.5 - 0.25\log_{10}(Chl)]$$

and $\alpha_l = 1 - \alpha_s$. Figure 9 shows phase functions determined by Eq. (11), along with the frequently used Petzold average-particle phase function.

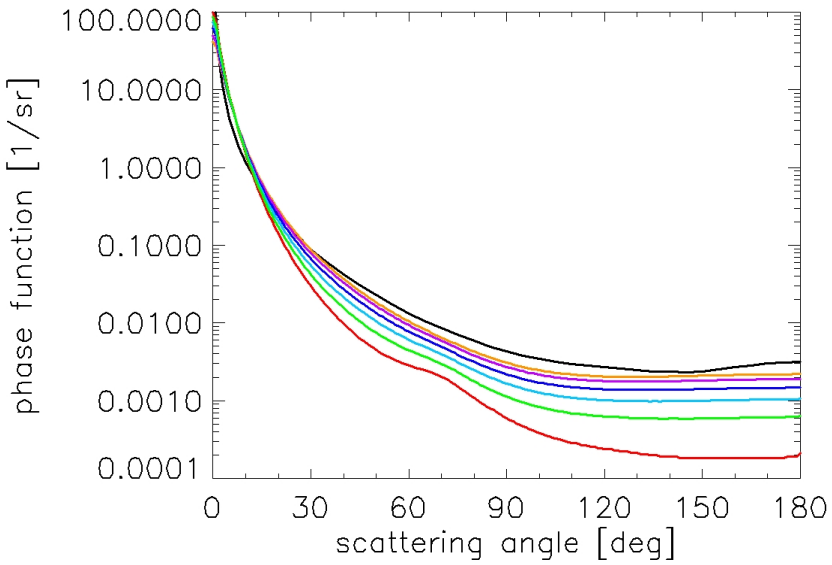

Fig. 9. Phase functions for small (orange) and large (red) particles as given by Morel et al. (2002). Phase functions as given by Eq. (11) for *Chl* = 0.01 (purple), 0.1 (blue), 1.0 (teal), and 10.0 (green), and the Petzold average particle phase function (black) are also shown.

It should be noted that the Morel et al. (2002) phase functions have smaller backscatter fractions ($B_p = 0.014$ for the small particles to 0.0019 for the large particles) than the Petzold phase function ($B_p = 0.018$). This is consistent with what is known about the phase functions for algal particles (e.g., Ulloa et al., 1994, or Twardowski et al, 2001; recall Eq. 5). The Morel phase functions of Eq. (11) and Fig. 9 are adopted for use in the new Case 1 IOP model.

All IOPs for Case 1 water are now determined by the user-specified chlorophyll profile (after selecting the particular of *A* and *B* spectra corresponding to low, medium, or high UV absorption).

To illustrate the quantitative differences (including the combined effects of absorption and scattering coefficients and the particle scattering phase function) between the classic and new Case 1 IOP models, Fig. 10 shows the remote-sensing reflectance $R_{rs}(\lambda)$ for homogeneous, infinitely deep waters with $Chl = 0.01$, 0.1, 1, and 10 mg m^{-3}, as computed for the classic and new Case 1 IOP models. The mid-range UV absorption model was used in the new model. The sun was placed at a zenith angle of 30 deg in a clear sky with typical marine atmospheric parameters (sky irradiances were computed using the RADTRAN-X sky irradiance model discussed in §5.1). The wind speed was 6 m/s. For the classic IOP model, the particle phase function was taken to be a Fournier-Forand phase function with a backscatter fraction as given by Eq. (5). EcoLight was run from 300 to 800 nm with 10 nm bands.

We see that the R_{rs} spectra are very similar for $Chl = 0.01$ and 0.1, but that the differences can become very large at high chlorophyll values. The maximum difference computed as 100(new - old)/old is less than 20% for Chl = 0.01 or 0.1. For $Chl = 1$, the maximum difference is less than 50% at visible wavelengths (58% at 795 nm). For $Chl = 10$, the differences are as large as 243% (more than a factor of three; at 575 nm). The larger R_{rs} for high *Chl* is due to the greatly increased scattering, as seen in Fig. 8.

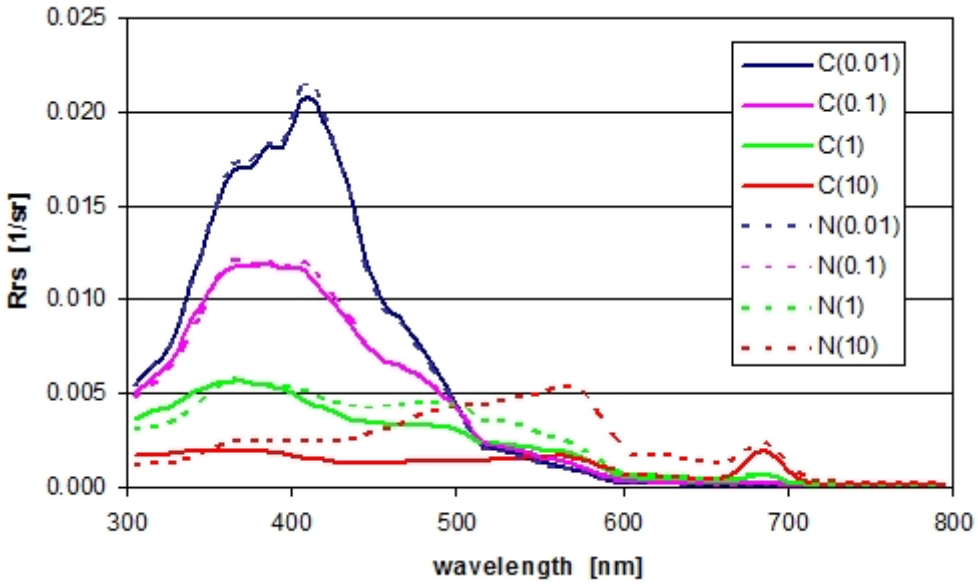

Fig 10. Comparison of R_{rs} as computed by the New Case 1 model with mid-range UV absorption (N; dotted lines) and the Classic model (C; solid lines), for *Chl* = 0.01, 0.1, 1, and 10 mg Chl m^{-3}.

We hope that this new IOP model for Case 1 water is an improvement over the classic model. However, the limitations of any IOP model must be remembered (in particular, see Mobley et al., 2004 for limitations of the "Case 1" concept). IOP models may be very good *on average*, but may or may not be (very often, *definitely are not*) correct in any particular instance. IOP models are best used for "generic" studies. When modeling a particular water body, especially in a closure study, it is always best to use measured data to the greatest extent possible. If, for example, measured particle absorption spectra are available, they can be used in HE52 by dividing the measured a_p by *Chl* to get a^*, which is then entered as A in a user defined AE file (with $E = 1$, as noted above). The corresponding *Chl* value can then be entered in the HE52 UI for use in reconstituting the measured a_p and for modeling the scattering (if not measured), as just described.

2.5 CASE 2. This model is a generic four-component (pure water, chlorophyll-bearing particles, CDOM, and mineral particles) IOP model. The UI allows users to specify individual component IOPs and concentration profiles in a variety of ways. Thus, the case 2 IOP model is suitable for many Case 2 waters. The various ways of specifying component absorption and scattering properties are similar to what has just been described in the Case 1 IOP models and are best seen by running the UI. This routine also gives the user additional options for modeling Case 1 waters (e.g., by specifying the microbial and CDOM components as desired, and setting the mineral concentration to zero). The various mass-specific absorption and scattering spectra available in HE52 are discussed in §2.8 and 2.9 below.

2.6 MEASURED IOP DATA. A common use of HE52 is to compute light fields using absorption *a* and beam attenuation *c* coefficients as measured by a WETLabs ac-9, ac-s, or equivalent instrument as input. The IOP model on file HE52\code\common\abacbb.f is designed for precisely this task. This IOP model divides the total absorption and scattering into two components: component 1 is pure water, and component 2 is "everything else," namely the particles and dissolved substances detected by the ac-9 (after pure water values are subtracted from the measured totals according to the recommended ac-9 data-processing protocol). For many (but not all) waters, the measured signal can be attributed primarily to particles, and the scattering for component 2 can be modeled with a particle-type phase function. The preparation process for including ac-9 data in an HE52 run is described below in §7.3.

This IOP model also has the option of reading in data from both unfiltered and filtered ac-9 instruments. The unfiltered ac-9 file gives the total (particles plus CDOM) absorption, and the filtered ac-9 file gives only the CDOM absorption. Having filtered and unfiltered ac-9s allows the total absorption to be separated into particulate and dissolved fractions, which gives important information about the ecosystem. However, it must be understood that light is influenced only by the *total* absorption coefficient, the *total* scattering coefficient, and the *total* phase function. If CDOM is assumed to be non-scattering, partitioning the absorption into particulate and dissolved fractions does not change the total *a*, total *b*, or total phase function. Therefore, partitioning the absorption into particulate and dissolved fractions does *not* change the elastic-scattering solution of the radiative transfer equation. Thus, in the absence of fluorescence, the light field is *exactly* the same whether or not the absorption is partitioned into particulate and dissolved components. Oceanographers may care about absorption by particles vs. absorption by CDOM, but photons do not.

The one optical situation for which partitioning absorption into particulate and dissolved components is necessary is in the CDOM fluorescence calculation. To predict CDOM fluorescence, it is first necessary to know how much light was absorbed by CDOM. Therefore, if the user selects the option of including CDOM fluorescence in the HE52 run, the user is given the option of naming a data file containing *a* and *c* values from a filtered ac-9. These CDOM absorption values are used *only* in the fluorescence calculations. In most instances, CDOM fluorescence is a small contribution to the total light field and has an almost negligible effect on quantities such as the water-leaving radiance.

An important additional feature of the Measured IOP data model is the option of reading a file containing backscatter coefficients $b_b(z,\lambda)$ as measured by a WETLabs bb-9, HOBILabs HydroScat-6, or similar instrument. If such data are available, HE52 can use the backscatter fraction $B(z,\lambda) = b_b(z,\lambda)/b(z,\lambda)$ to generate a phase function having the measured backscatter fraction at each depth and wavelength. Here, the $b(z,\lambda)$ values are those obtained from the ac-9 data. If this option is used, the user does not select a phase function for component 2 from within the UI.

When processing *a* and *c* data from an ac-9 or ac-s, the convention is to remove the contribution by pure water to the measured total. HE52 therefore assumes that standard-format files of *a* and *c* data have water removed. The chosen pure water values are then automatically added back in to create the total IOPs needed to solve the RTE. However, there is no convention on removing pure water backscatter values when processing bb-9 or HydroScat-6 data. When reading a standard-format file of backscatter data, the user must indicate on the UI whether the b_b data do or do not include backscatter by pure water.

When $b_b(z,\lambda)$ and $b(z,\lambda)$ data are used to dynamically determine the phase function according to the backscatter fraction, a Fournier-Forand (Fournier and Jonasz, 1999; c.f., Fournier and Forand, 1994) phase function is used. This closed-form, analytical phase function is based on Mie theory and is parameterized by the real index of refraction of the particles and the slope of the Junge size distribution. By varying the particle index of refraction and size distribution, phase functions can be generated with very small to very large backscatter fractions. A collection of discretized Fournier-Forand phase functions with backscatter fractions from 0.0001 to 0.5 is built into HE52. Figure 11 shows selected phase functions from this collection. When the option to dynamically determine the phase function from the backscatter fraction is chosen, HE52 interpolates within this collection to generate a phase function at each depth and wavelength with the needed amount of backscatter. This interpolation requires very little increase in run time compared to using the same phase function for all depths and wavelengths.

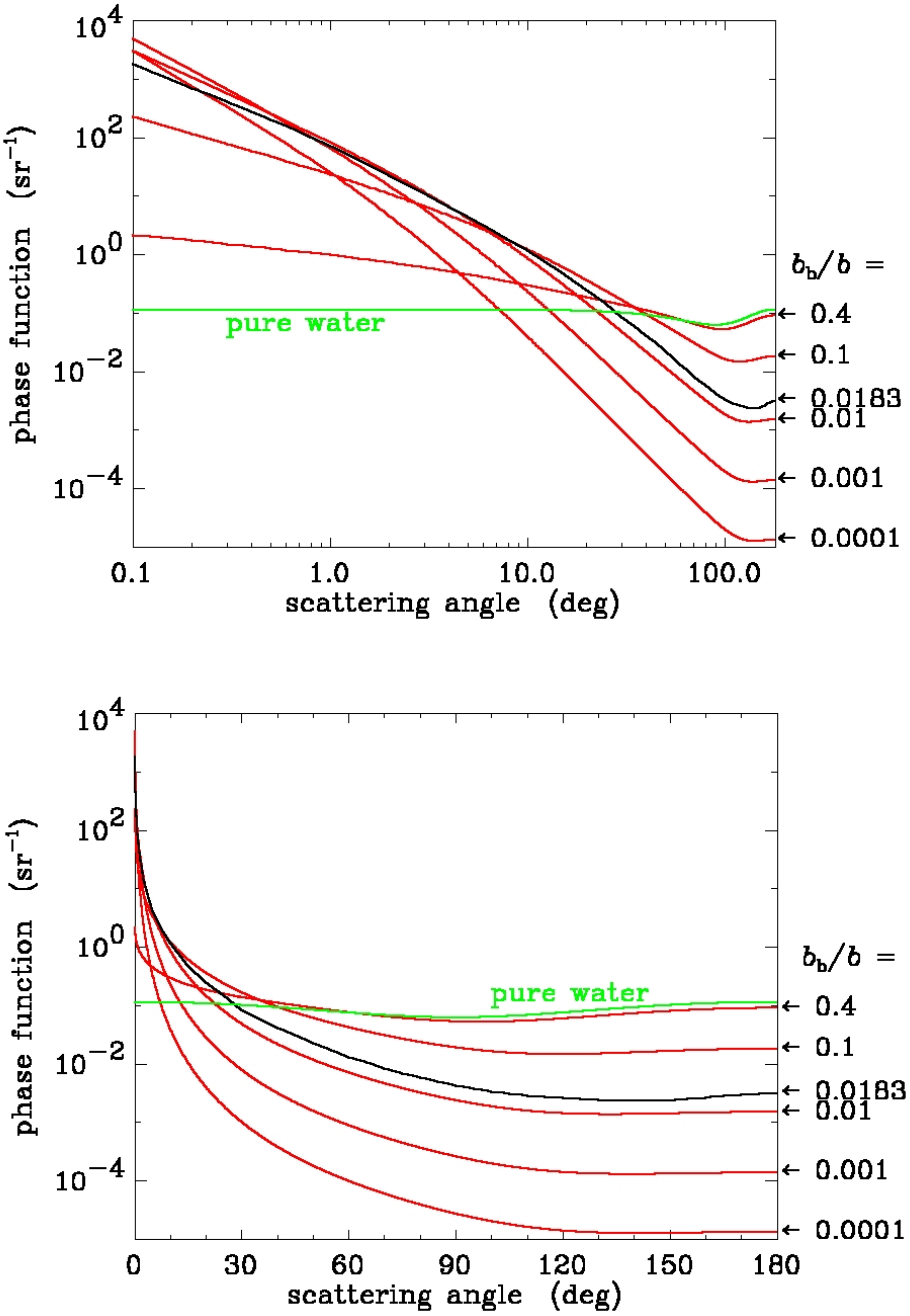

Figure 11. Selected Fournier-Forand phase functions (red lines). The black line labeled b_b/b = 0.0183 is the Petzold average-particle phase function (on file HE52\data\phasefun\...\avgpart.dpf). The pure water phase function (green line, file pureh2o.dpf) has $b_b/b = 0.5$.

The Fournier-Forand phase functions used for dynamic determination of the phase function also can be selected "by name" in the UI, just like any other phase function. If this is done, the named phase function will be used at all depths and wavelengths for that IOP component. The Fournier-Forand phase functions are named by their backscatter fraction. For example, the phase function on file FFbb0001.dpf has a backscatter fraction of 0.0001; phase function FFbb018.dpf has a backscatter fraction of 0.018, and so on. However, it is just as easy to specific the backscatter fraction, e.g., 0.018, in which case the Fournier-Forand phase function on file FFbb018.dpf will be used.

The .dpf extension of the phase function files just means that the file contains a discretized phase function. The files themselves are ASCII text files. However, it is not meaningful to open a *.dpf file and look at the contents in hopes of seeing the phase function as a function of scattering angle. The discretized phase functions used by HydroLight are the quad-averaged phase functions computed as described in *Light and Water*, Chapter 8. This computation is carried out by the code in the HE52\SpecialRuns\PhaseFunction directory, which is described in §9.2 below.

A couple of final comments are in order regarding the use of Measured IOPdata. First, **it is not HE52's job to quality control your ac-9, ac-s, bb-9, HydroScat-6, or other data. HE52 usually assumes that your data are perfect, and it gives you the corresponding solution of the radiative transfer equation** (recall Fig. 1 of the Users' Guide). One exception to this is that if HE52 detects a negative absorption coefficient in your data file (a common occurrence due to instrument noise at near-infrared wavelengths where phytoplankton absorption is essentially zero), it will print a warning message to the printout file and reset the negative absorption to zero, and then continue running. HE52 will often continue running, even with obviously bad input. It is not idiot proof and does not do much checking of user input data. Second, if you are using ac-9 data to define the IOPs and you also wish to include chlorophyll fluorescence, for example, the UI will inform you that chlorophyll is not a part of your IOP model and ask you to input information on the chlorophyll profile. But, you say, "The reason I bought an ac-9 was to measure the IOPs directly, rather than having to measure the chlorophyll profile and then use a bio-optical model to get the IOPs." That's fine, but you still need to know how much chlorophyll is in the water before you can compute the chlorophyll fluorescence.

There is an additional caveat when using WETLabs ac-s data. The HydroLight Standard Format for files of a and c data assumes that both a and c are measured at the same wavelengths. This is true for the ac-9 instrument. However, the ac-s instrument measures

the same number of wavelengths for both *a* and *c*, but the exact wavelengths for each can differ by a few nanometers. Therefore, **when working with ac-s data, the user should spline the *a* and *c* spectra to a common set of wavelengths, say at 5 nm resolution, and use those data to create a standard-format data file for input to HE52.**

2.7 USER IOP MODELS. This option on the UI gives users convenient access to their own IOP models from within the HE52 UI. If this IOP model is chosen on the UI, the user is asked to enter the name of the subroutine containing the user's IOP model and to enter a name for each component of the user's IOP model. These component names are then used by the UI to identify the components, whose IOPs can be specified in a manner similar to that for the case2 model. Note that, unlike the other IOP models, there is no corresponding "ab" file in the HE52\code\common directory, because you, the user, must write the desired subroutine defining your IOP model. Section 8.1 describes how to write IOP subroutines, which can then be used with the other UI option. **Note that user-written IOP (and any other user-written) subroutines must be placed in the HE52\code\user directory.**

The HE52 code as distributed contains an example of a user-written IOP model on file TenCompIOPs.f in the HE52\code\user directory. The model builds up the total IOPs from 10 components, which in the original application were pure water, CDOM, minerals, and 7 phytoplankton species. This model can be used as is, or can serve as a template for modifications to define other IOPs as needed by a particular user. Note that when running any IOP model with more than 4 components (the number needed for the Case 2 IOP model), the maximum number of IOP components must be increased on the UI Change Limits form.

2.8 Specific-Absorption Models

Some of the IOP models require a chlorophyll or mineral particle concentration to be converted to absorption and scattering coefficients. Such conversions are made by equations of the form

$$a(z,\lambda) = a^*(\lambda)\, X(z)\,.$$

The absorption coefficient a is always in units of m^{-1}, a^* is the chlorophyll- or mass-specific absorption coefficient, and X is the component concentration. **For chlorophyll, a^* is in m^2 (mg Chl)$^{-1}$, and X is the chlorophyll concentration in (mg Chl) m^{-3}. For mineral**

particles, a^* is in m^2 $(g)^{-1}$, and X is the mineral particle concentration in g m^{-3}. HE52 uses these default units because they are what is standard in the literature when expressing concentrations of chlorophyll and mineral particles. HE52 comes with several examples of component-specific absorption and scattering spectra. The user's a^* and X values can, however, be in any units so long as the product of mass-specific absorption times concentration has units of inverse meters (e.g., a^* in m^2 (µg Chl)$^{-1}$ and X (µg Chl) m^{-3}). The user can easily add additional spectra.

The file HE52\data\defaults\apstarchl.txt contains a default chlorophyll-specific absorption coefficient, which is shown in Fig. 12. This particular spectrum is taken from Morel (1988, his Fig. 10c, which is Prieur and Sathyendranath, 1981, Table 2, Column 3), scaled so that $a^*(440nm) = 0.05$ m^2 (mg Chl)$^{-1}$, which is a typical value for $a^*(440nm)$. This spectrum was selected as the default because its normalized version [with $a^*(440nm) = 1$] shown in Fig. 2 is used in the classic case1 IOP model. Note that there is at least a factor of five difference among measured values of specific absorption (e.g., Prieur and Sathyendranath reported values between 0.02 and 0.1 m^2 (mg Chl)$^{-1}$ at 440 nm), so the values in the default file may or may not describe your actual water body.

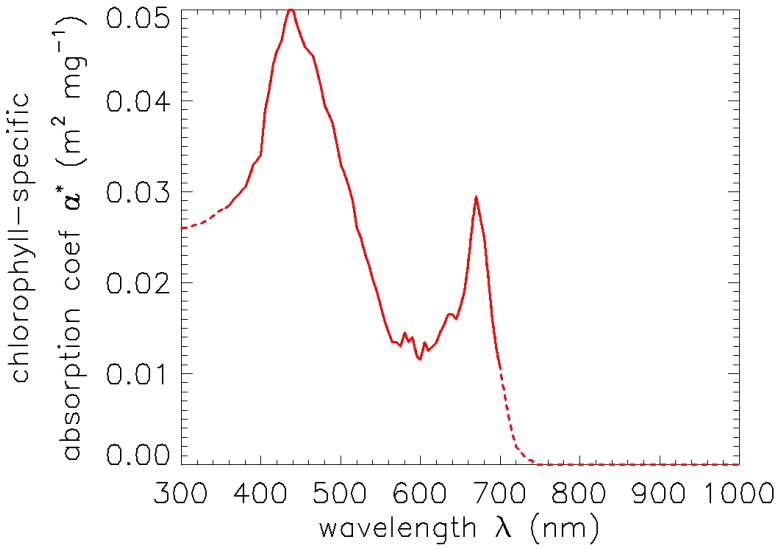

Figure 12. The default chlorophyll-specific absorption spectrum on file apstarchl.txt. This spectrum is based on Prieur and Sathyendranath (1981).

Several mass-specific absorption spectra for mineral particles are included in the HE52\data\defaults directory. Files containing values taken from Fig 4-3A of Ahn (1999) are available for brown earth, calcareous sand, yellow clay, red clay, and the average of these four. These files are named astarmin_X.txt where X labels the type of mineral particle in the file. Ahn measured these spectra in the laboratory for 400-750 nm. These curves were extrapolated by eye to 300 and 1000 nm for use in HE52. **The spectra are therefore very uncertain, and perhaps simply wrong, near 300 and 1000 nm.** These specific absorption curves are shown in Fig. 13. As noted above, the mineral particle mass-specific absorption spectra have units of $m^2\ g^{-1}$; thus the mineral concentrations entered via the UI must be specified in $g\ m^{-3}$. These data files are all provided with HE52 as *examples* of what is available in the literature; no statement is made about their accuracy or applicability to your particular water body. Users wishing to use their own chlorophyll- or mineral-specific absorption spectra can create files on the format described in §7.2, and then select those files from within the UI.

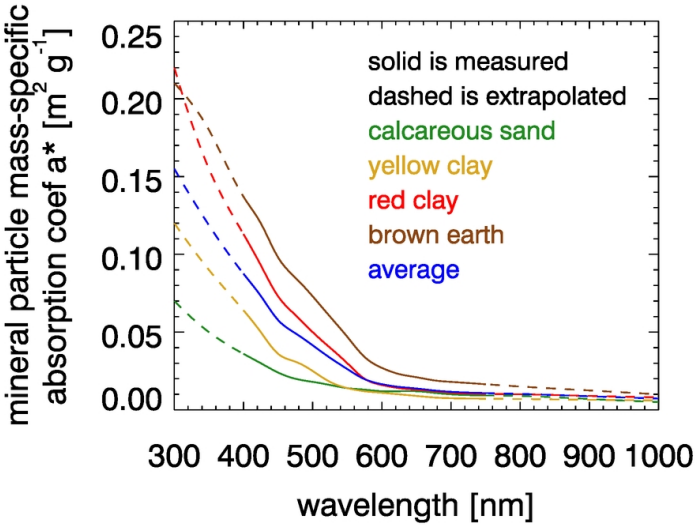

Figure 13. Mineral mass-specific absorption spectra as provided with HE52.

CDOM absorption is usually modeled as an exponentially decaying function of wavelength:

$$a_{CDOM}(z,\lambda) = a_{CDOM}(z,\lambda_0)\exp[-\gamma(\lambda - \lambda_0)]\,.$$

The model parameters λ_o and γ can be specified in the UI. Various options are available for specifying $a_{CDOM}(z,\lambda_o)$. One option is to let $a_{CDOM}(z,\lambda_o)$ covary with the chlorophyll absorption, as it does in the classic case1 IOP model. Then

$$a_{CDOM}(z,\lambda) = 0.2\, a_{chl}(z,440) \exp[-0.014(\lambda - 440)] .$$

It is also possible to read a user-supplied subroutine or data file to obtain $a_{CDOM}(z,\lambda_o)$.

2.9 Specific-Scattering Models

The UI provides several options for how to model scattering by IOP components. Scattering is often modeled by a power-law dependence on wavelength,

$$b(z,\lambda) = b_0 \left(\frac{\lambda_o}{\lambda} \right)^m X(z)^n ,$$

where X is a concentration (e.g., Chl in mg m^{-3} or mineral particles in gm m^{-3}), and b_o, λ_o, m, and n are model parameters. The commonly used Gordon-Morel (1983) scattering model for chlorophyll-bearing particles is a special case of the power-law model with $X = Chl$, $b_o = 0.3$, $\lambda_o = 550$ nm, $m = 1$, and $n = 0.62$:

$$b_{chl}(z,\lambda) = 0.3 \left(\frac{550}{\lambda} \right) [Chl(z)]^{0.62} .$$

This model for the scattering coefficient is used in the classic case1 IOP model.

Gould, *et al.* (1999) presented a scattering model that is a linear function of wavelength:

$$b(z,\lambda) = b_0 \left(\frac{m\lambda + i}{m\lambda_0 + i} \right) X(z)^n .$$

Their best-fit parameters for a range of water bodies are $b_o = 0.5$, $m = -0.00113$, $\lambda_o = 550$ nm, $i = 1.62517$, and $n = 0.62$. This model can be selected in the UI when using the Case 2 IOP model. Note that these two power-law and linear models are models of the scattering coefficient rather than models of the concentration-specific scattering coefficient.

Scattering can also be modeled by specifying a file of specific scattering data and a file of concentration data:

$$b(z,\lambda) = b^*(\lambda)\, X(z)\,,$$

where b^* is the concentration-specific scattering coefficient and X is the concentration.

Files of mass-specific scattering data for mineral particles corresponding to the mass-specific absorption data seen in Fig. 13 are also available in the HE52\data\defaults directory. Figure 14 shows the wavelength dependence of these mass-specific scattering coefficients given by the various data files. Again, users are free to use either the provided models or their own data files to specify the specific scattering coefficient for each component. If using the example mineral spectra provided with HE52, note that HE52 does not check to see if you have chosen self-consistent mineral a^* and b^* spectra, e.g., that you have chosen b^* for red clay if you picked a^* for red clay.

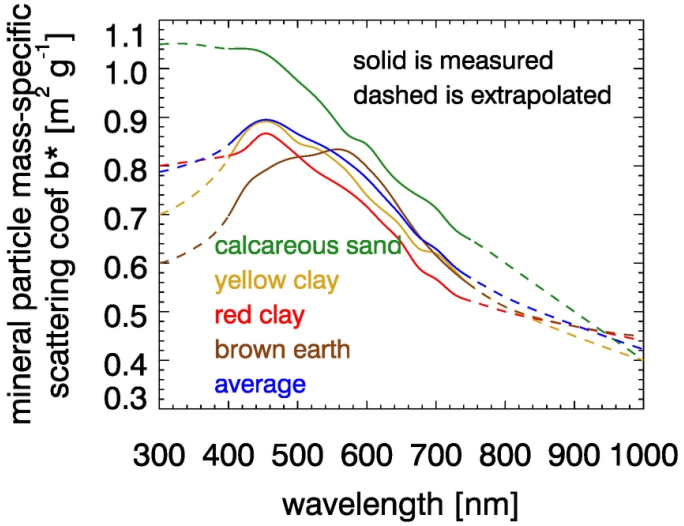

Figure 14. Mineral mass-specific scattering spectra as provided with HE52.

3. Bottom Reflectance Models

3.1 Finite-depth Water

HE52 can simulate both finite-depth and infinitely deep water bodies. In the finite-depth case, a physical bottom is placed at depth z_{max}, where z_{max} is the last depth specified in the list of depths where output is requested in the UI. The default is to assume that the physical bottom is an opaque, Lambertian reflecting surface whose *irradiance* reflectance $R = E_u / E_d$ is specified by the user. This irradiance reflectance is automatically combined with a Lambertian bi-directional reflectance distribution function (BRDF) to generate the needed *radiance* reflectance properties of the bottom. [Equation (4.81) of *Light and Water* gives the needed bi-directional radiance reflectance (BRRF) of the physical bottom; the BRRF equals the BRDF times the cosine of the incident polar angle.]

Data files containing irradiance reflectances for various sediments and benthic biota are provided with HE52. These files include are based on measurements from 400 to 750 or 800 nm, which were extrapolated by eye to 300 and 1000 nm for use in HE52. Thus the reflectances may be unrealistic near 300 and 1000 nm. These reflectance spectra are shown in Fig. 15. As with all other such data sets, these spectra are provided with HE52 as *examples* of bottom reflectances; they may or may not be adequate for modeling your water body. Section 7.5 describes how to add your own data files to the list of available bottom reflectance spectra.

The source code for the default Lambertian bottom BRDF is contained on file HE52\code\Hydrolight\BRDFLamb.f. The assumption of a Lambertian bottom is justified for most simulations. However, HydroLight can simulate non-Lambertian bottoms (e.g., Mobley et al., 2003). Section 8.4 describes how to write HydroLight subroutines for non-Lambertian bottoms. EcoLight always assumes that a finite-depth bottom is Lambertian.

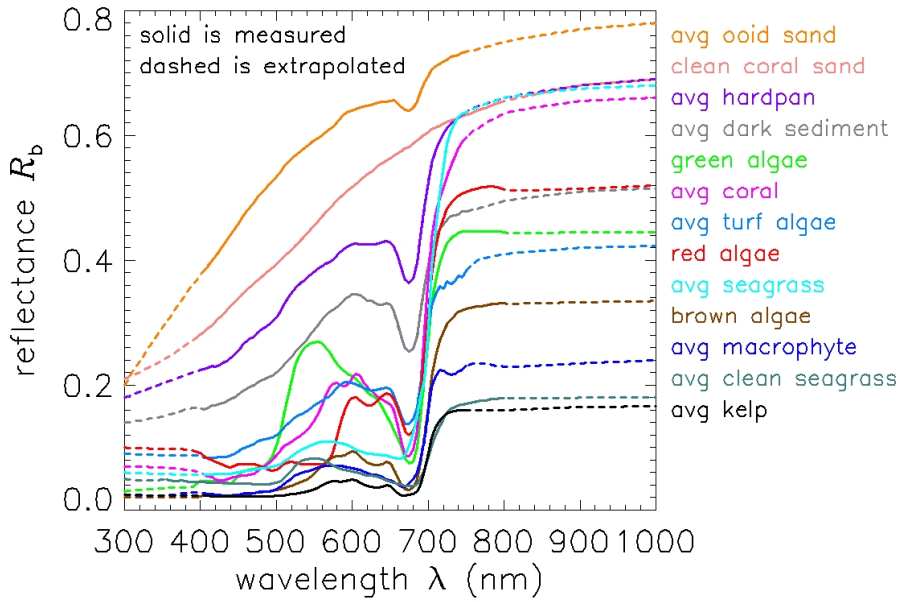

Figure 15. Bottom irradiance reflectances as provided with HE52. Solid lines are measured; dashed lines are subjective extrapolations.

3.2 Infinitely Deep Water Without Inelastic Scattering or Internal Sources

The complicated situation for infinitely deep water requires explanation. Consider first the case of no inelastic scattering (no Raman scattering, or chlorophyll or CDOM fluorescence) and no internal sources (no bioluminescence). This is called the "source-free" case. Now z_{max} gives the "maximum output depth" for the run, i.e., the maximum depth at which we wish to obtain output from HE52. For a study in the open ocean, we might take z_{max} = 50 m, even though the water is optically infinitely deep. In this case, **HE52 always assumes that the water is homogeneous below depth z_{max} and has the same IOPs as are given by the IOP routine at $z = z_{max}$.** Note that the water column between depth 0 (the mean sea surface) and depth z_{max} generally has depth-dependent IOP's. For infinitely deep water, HE52 automatically computes the BRDF of the infinitely deep, source-free, homogeneous layer of water below depth z_{max} and applies this BRDF as the bottom boundary condition *at* depth z_{max}. (Section 9.5 of *Light and Water* describes these calculations as performed by HydroLight.) The infinitely deep layer of water below depth z_{max} is not a Lambertian reflector.

Because the water column is assumed to be homogeneous below depth z_{max}, different choices for z_{max} can give greatly different light fields between depths 0 and z_{max} if the water IOPs vary strongly with depth. This is illustrated in Fig.16, which shows a true IOP profile (e.g., the absorption coefficient at a particular wavelength) for the upper 30 m of a water column, and the IOP profiles that will be used in HE52 if z_{max} is set to 10 and 20 m. Clearly, these IOP profiles represent different radiative transfer problems and the light fields will be different in the regions above z_{max} for these three situations. **When modeling infinitely deep water with depth-dependent IOPs, the user must always take z_{max} to be far enough below the maximum depth of interest that the light field in the region of interest is not significantly affected by light from below z_{max}.** This depth often must be determined by a few trial and error runs.

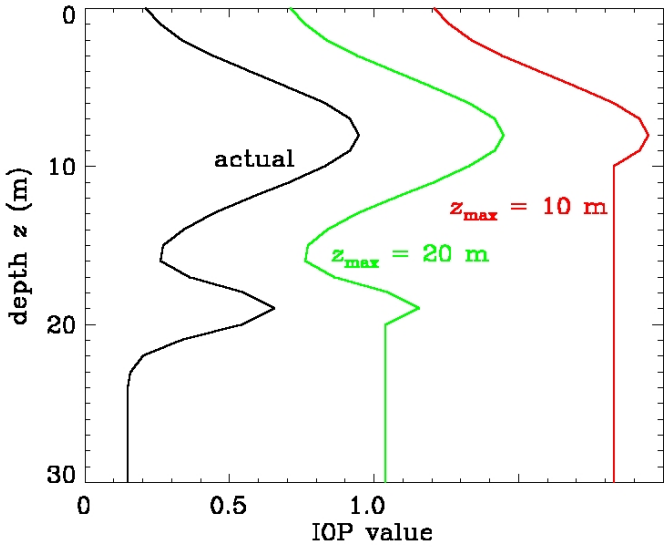

Figure 16. Hypothetical actual depth profile of an IOP (black line) and the profiles used when z_{max} = 10 (red) and 20 m (green). The z_{max} = 20 m profile is offset by 0.5 to the right for clarity; the z_{max} = 10 m profile is offset by 1.0.

3.3 Infinitely Deep Water with Inelastic Scattering or Internal Sources—Dynamic Bottom Boundary Depths

For technical radiative transfer reasons (well, in all honesty, because Curt can't figure out the math), the water column below depth z_{max} does not include inelastic scattering or bioluminescence, even if those effects are included between the surface and z_{max}. In other words, inelastic scattering and bioluminescence are always "turned off" at the lower boundary. This can cause abrupt changes in diffuse attenuation functions near z_{max} and other apparently strange behavior as the light field near z_{max} adjusts from one including inelastic scatter above z_{max} to one without it below z_{max}. These behaviors are illustrated and discussed in detail in HydroLight Technical Note 6, which is available on the HE52\documents directory or the HydroLight Users' Group website.

To avoid problems or confusion with the application of a source-free bottom BRDF in runs involving internal sources and infinitely deep water, HE52 by default applies the bottom boundary condition at a wavelength-dependent, dynamically determined (determined at run time) depth below the greatest output depth requested by the user. This dynamic bottom boundary depth guarantees that any effects due to the use of a source-free lower boundary condition in a run including internal sources will not affect the radiances and other quantities at the greatest depth of interest.

Runs for a wide range of water conditions from very clear to very turbid, and for wavelengths from 300 to 1000 nm, show that applying the bottom boundary condition at 20 optical depths below the greatest depth of interest is sufficient to give good radiances at the greatest depth of interest. HE52 therefore does the following:

At the start of a run it computes the wavelength-dependent depth

$$z_b(\lambda) = z_{max}(\text{requested}) + 20/c(z_{max}, \lambda),$$

where $c(z_{max}, \lambda)$ is the beam-attenuation coefficient at the maximum output depth requested by the user. It then finds $z_b(\lambda_{max})$, the maximum of $z_b(\lambda)$, which occurs at some wavelength λ_{max}. The infinitely deep, source-free bottom boundary condition is then applied at depth

$$z_b(\lambda) = z_b(\lambda_{max}) \text{ if } \lambda \le \lambda_{max}$$

$$z_b(\lambda) = z_{max}(\text{requested}) + 20/c(z_{max}, \lambda) \text{ if } \lambda > \lambda_{max}$$

These bottom depths guarantee that the RTE is solved as deeply as needed at all wavelengths $\lambda < \lambda_{max}$ in order to compute the inelastic scattering source terms needed at wavelength λ_{max}, where the water is clearest, and that it is solved no deeper than needed at longer wavelengths (especially beyond 700 nm) where water absorption is high.

Use of the dynamic bottom boundary depth is the default in HE52 whenever bioluminescence or inelastic scatter are included and the water column is infinitely deep. If no source terms are included or the bottom is placed at a finite depth, then the bottom boundary condition is applied at z_{max}(requested), just as in previous versions of HydroLight. If desired, the dynamic bottom boundary option can be turned off in the "Change Defaults" form of the HE52 User Interface. In order to avoid a new type of confusion, output at depths below the user-requested maximum depth z_{max}(requested) is not included in the various output files. Ignorance is bliss.

Experience shows that the dynamic bottom boundary condition typically adds about 20% to the run time, compared to the use of a fixed bottom depth. That is a reasonable computational price to pay in order to avoid spurious results in simulations where the effects of sources are of interest at great depths. The penalty for not going deep enough can be an apparently "wrong" answer for your problem.

4. The Sea Surface Model

HE52 models the wind-blown sea surface using Cox-Munk (1954a, 1954b) wind-speed wave-slope statistics, temperature- and salinity-dependent Fresnel reflectances, and Monte Carlo ray tracing as described in *Light and Water* Chapter 4. It should be noted that the Cox-Munk slope statistics are based on observations of the ocean surface and therefore include the effects of whatever gravity and capillary waves where on the sea surface at the time of their observations. The Cox-Munk statistics are sometimes thought of as describing the slopes of capillary waves. In some situations this viewpoint is reasonable because much of the variance in the slope of the sea surface is indeed due to the smallest capillary waves. The larger gravity waves are responsible for most of the variance of the sea surface elevation, but they contribute less to the variance of the surface slope. These matters are discussed in more detail in HydroLight Technical Note 1, which can be found in the HE52\documents directory.

Hydrolight uses Monte Carlo ray tracing of millions of rays (or photon packets) through tens of thousands of randomly generated sea surface realizations to compute the radiance reflectance and transmittance functions that describe the optical properties of the sea surface (see *L&W* §4.7). When doing Monte Carlo simulations of rays interacting with the sea surface, it is the slope of the surface at the point where a ray intersects it that determines the directions of the reflected and transmitted rays. The magnitudes of the transmitted and reflected rays are determined by the associated Fresnel reflectance and transmittance, which are determined by the real index of refraction of the water. Thus if the slope statistics and water index of refraction are correct, then the optical properties of the surface will be accurately modeled in most situations. The exception may occur if the sun or viewing direction is near the horizon, when "wave shadowing" by large gravity waves may be significant.

It should be noted that the Monte Carlo ray tracing that computes the surface reflectances and transmittances exactly conserves energy as light passes through the sea surface. However, conservation of energy across a wind-blown sea surface has some radiometric subtleties when expressed in terms of radiance or irradiance. These matters are discussed in HydroLight Technical Note 7 found in the HE52\documents directory.

The sea surface slope depends on the wind speed. The Fresnel reflectance of an air-water surface is determined by the real index of refraction n of the water, which in turn depends on the water temperature and salinity. The index of refraction varies over visible wavelengths (400-700 nm) by only a few percent about a value of $n = 1.34$. This value was therefore

used to generate the sea surface reflectances and transmittances in previous versions of HydroLight. Those sea surface reflectances and transmittances were then functions only of wind speed. However, the dependence of n on wavelength, temperature, and salinity becomes greater at UV and IR wavelengths, as seen in Fig. 17. HE52 therefore models the sea surface reflectance and transmittance properties as functions of wind speed, temperature, salinity, and wavelength.

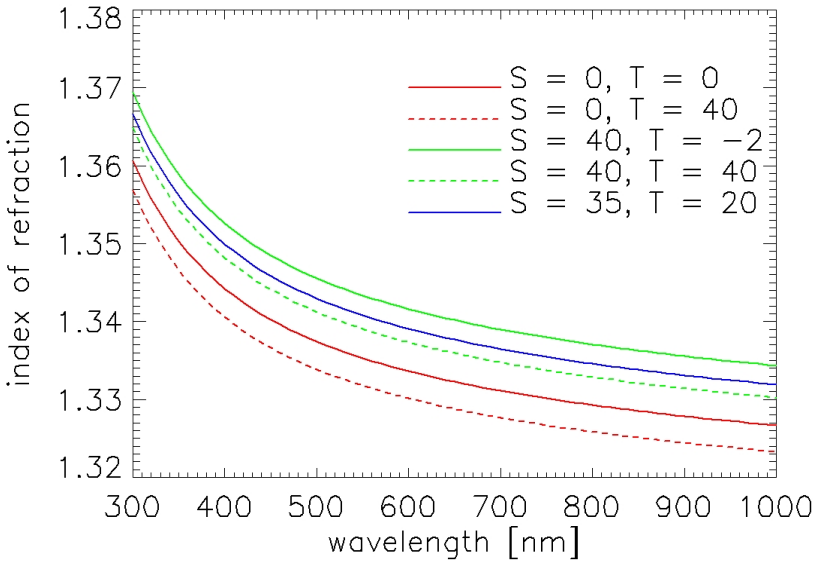

Fig. 17. Real index of refraction of water as a function of salinity S (psu), temperature T (deg C), and wavelength. Values are computed by Eq. (3) of Quan and Fry (1995).

The Monte Carlo surface calculations are performed as a "special run" using code described in §9.1. These calculations are time consuming, but they need be done only once for each combination of index of refraction and wind speed. The results are stored (the files in directory HE52\data\surfaces) as a function of index of refraction from $n = 1.32$ to 1.38 and wind speeds of $U = 0$ to 15 m s^{-1}. When HE52 runs it then uses the formula of Quan and Fry (1995; their Eq. 3) to convert the salinity and temperature (as specified by the user in the UI) to n at each wavelength. The available surface files are then interpolated to the needed value of n and U at each wavelength.

Temperature and salinity effects on the Fresnel reflectance cause less than a 5% variability in remote-sensing reflectance near 300 nm, and less than 3% at visible wavelengths, compared to the use of a value of $n = 1.34$ at all wavelengths as in previous versions of HydroLight. However, some users have requested the ability to include salinity and temperature effects, so this feature was added to version 5.

5. Sky Models

HE52 must know the sky radiance incident onto the sea surface from all directions and for all wavelengths included in the HE52 run to be made. To increase flexibility in using modeled or measured sky radiances, this information is provided by two independent subroutines. The first subroutine returns the direct and diffuse components of the sky downwelling plane *irradiance*. These irradiances are used to set the magnitude of the sky *radiance*. The second subroutine returns the *angular pattern* of the sky radiance distribution. (This radiance pattern is integrated over direction to compute a sky irradiance, which is then forced to equal the irradiance given by the irradiance subroutine.) Together, these two routines yield a sky radiance distribution with the desired magnitude in all directions. HE52 comes with two versions of each routine, as shown in Table 2, which we now discuss.

file name	**type**	**user interface option**	**comments**
radtranx.f	irradiances	SEMI-EMPIRICAL SKY MODEL	Computes irradiances using RADTRAN-X. Recommended for general use.
hcnrad.f	radiance	SEMI-EMPIRICAL SKY MODEL	Computes the normalized radiance pattern using Harrison and Coombes (1988). Recommended for general use.
cosirrad.f	irradiances	IDEALIZED SKY MODEL	Uses user-supplied input from the UI to compute the irradiances. For idealized, single-wavelength runs.
cosrad.f	radiances	IDEALIZED SKY MODEL	Computes radiance patterns for uniform or heavy overcast skies by *L&W*, Eq. (4.50). For idealized, single-wavelength runs.

Table 2. The sky irradiance and radiance models distributed with HE52. The user interface option column shows which routines are called when either of the sky model options is selected when running the UI.

5.1 The RADTRAN-X Sky Irradiance Model

A major part of the work involved in extending the HE52 databases to the 300-1000 nm range required extending the original RADTRAN sky irradiance model of Gregg and Carder (1990), which is the default used to compute the clear-sky direct (sun) and diffuse (background sky) irradiances incident onto the sea surface. Dr. Marcos Montes of the U. S. Naval Research Laboratory kindly computed and provided the needed O_2, O_3, and H_2O absorption coefficients on the format needed to extend the RADTRAN model used in H4. The RADTRAN numerical model with the extended underlying database as used in HE52 is called RADTRAN-X.

At wavelengths near 300 nm the solar irradiance reaching the sea surface is strongly influenced by ozone absorption. It is therefore important that the user provides HE52 with the correct O_3 concentration, if accurate radiometric quantities are to be computed below ~340 nm. (AOPs will of course be insensitive to the magnitude of the incident sky irradiance.) To assist with this, HE52 includes a climatology of monthly averaged O_3 concentrations for 5 degree latitude by 10 deg longitude bins. This climatology was derived from 5 years (2000-2004) of TOMS (Total Ozone Mapping Spectrometer) version 8 satellite data. For polar regions with missing TOMS data, zonal averages from the nearest latitude band with valid data are used. This database is used when the user specifies the HE52 run for a particular latitude, longitude, and time, and specifies the use of climatological ozone values (by setting the ozone concentration to -99 in the UI). As before, a particular ozone value can be specified in Dobson units in the UI, which will override the climatological value. It should be noted that the ozone concentration affects the incident irradiance only below about 340 nm. Although the difference in E_d at the sea surface for very low and very high O3 concentrations can be an order of magnitude at 300 nm, there is no effect on AOPs.

Figure 18 shows an example of the TOMS monthly average climatological ozone values as available in HE52, and the actual values for a particular day. The monthly averaged values vary over the course of a year and with location from less than 150 to more than 450 Dobson units. The default used in HE52 is 300 Dobson units.

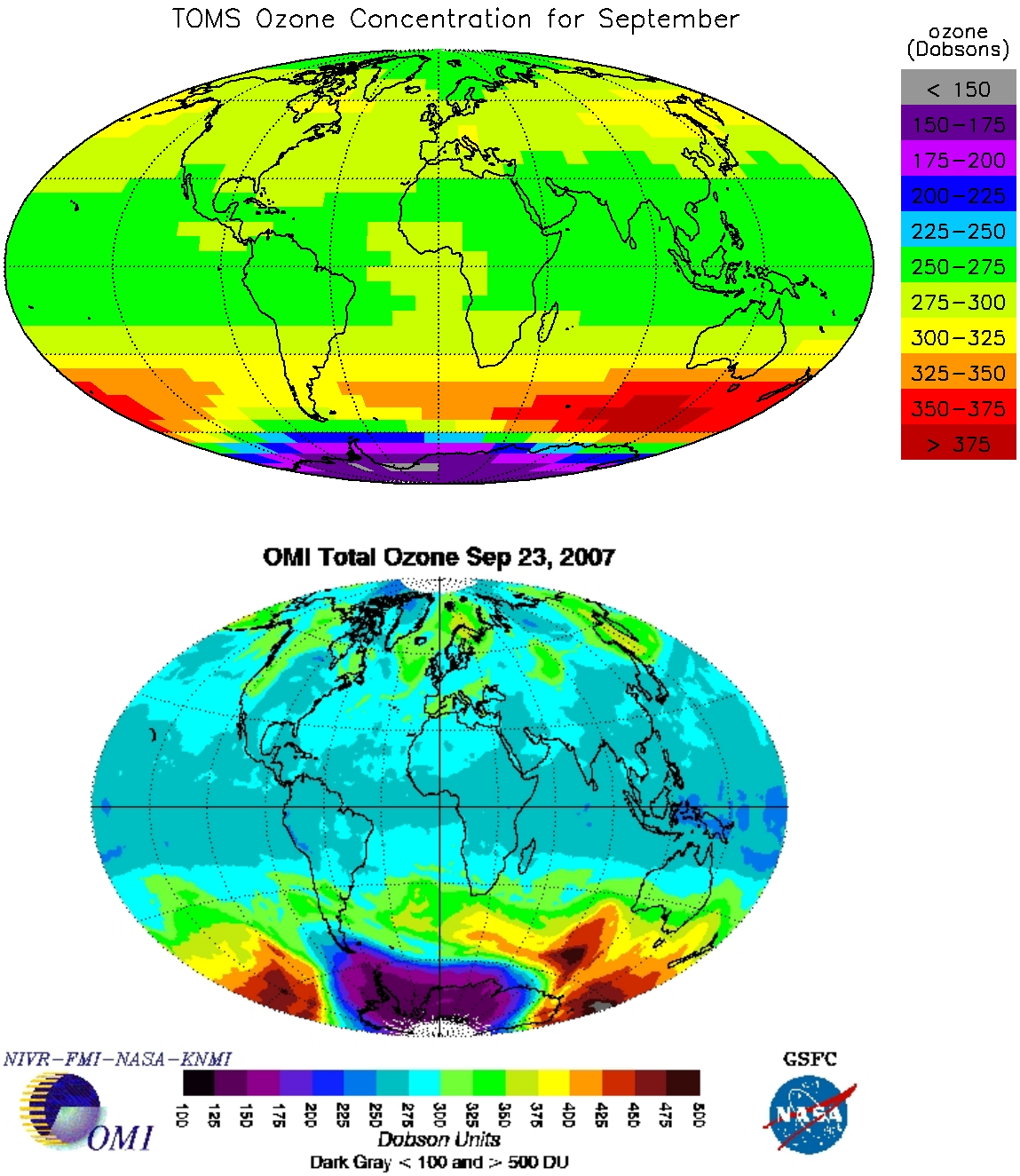

Fig. 18. Top: The TOMS monthly average ozone concentration for September, as used in HE52. Bottom: The measured ozone concentrations for a single day, 23 Sept. 2007 (from http://toms.gsfc.nasa.gov/ozone/ozone.html)

Figure 19 shows the total, direct (sun), and diffuse (sky) downwelling plane irradiances E_d at the sea surface for clear sky, typical open ocean atmospheric conditions, the sun at a 45 deg zenith angle, and a typical ozone concentration of 300 Dobson units. RADTRAN-X always outputs its irradiances at 1 nm resolution. HE52 automatically averages those irradiances over the wavelength bands requested in the run.

Figure 20 shows the effect of varying ozone concentrations near 300 nm for the same general conditions as Fig. 19. Although E_d at 300 nm is an order of magnitude higher for low ozone (150) than for high (450), ozone significantly affects the irradiances only below about 330 nm. For runs at 350 nm or greater, the value of the ozone concentration used in the HE52 run is irrelevant.

The RADTRAN-X clear-sky direct and diffuse sky irradiances are modified as described in Kasten and Czeplak (1980) if the cloud fraction *Cld* is greater than 0.25 ($0 \le Cld \le 1$):

$$E_{\mathrm{d}}(\mathrm{total},Cld) = E_{\mathrm{d}}(\mathrm{total},\mathrm{clear})\,[1 - 0.75\,Cld^{3.4}]\,,$$

and

$$E_{\mathrm{d}}(\mathrm{diffuse},Cld) = E_{\mathrm{d}}(\mathrm{total},Cld)\,[0.3 + 0.7\,Cld^{2}]\,.$$

Here $E_{\mathrm{d}}(\mathrm{total}) = E_{\mathrm{d}}(\mathrm{direct}) + E_{\mathrm{d}}(\mathrm{diffuse})$. Figure 21 shows the clear sky irradiances of Fig. 19 along with the irradiances for a 50 per cent cloud cover, with both averaged over 10 nm wavelength bands.

The Kasten and Czeplak adjustment for cloud fraction is very simple and cannot account for differences such as a thin cloud cover vs. part of the sky covered by small cumulus clouds, although both situations might correspond to a particular value of *Cld* in their model. Their model is, nevertheless, better than nothing in estimating the effects of clouds on incident irradiances.

If very accurate irradiance calculations are to be made, especially for cloudy conditions, then the UI has an option for specifying a user-created file of measured total irradiances, rather than using values computed by RADTRAN-X. If this is done, HE52 still uses RADTRAN-X to partition the measured total into direct and diffuse parts, as is needed within HE52.

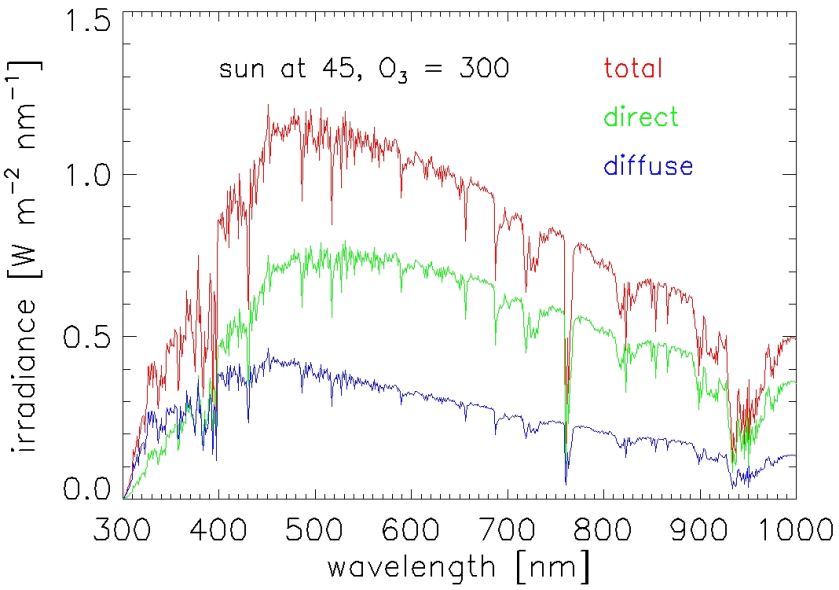

Fig. 19. Example sea surface irradiances as computed by RADTRAN-X.

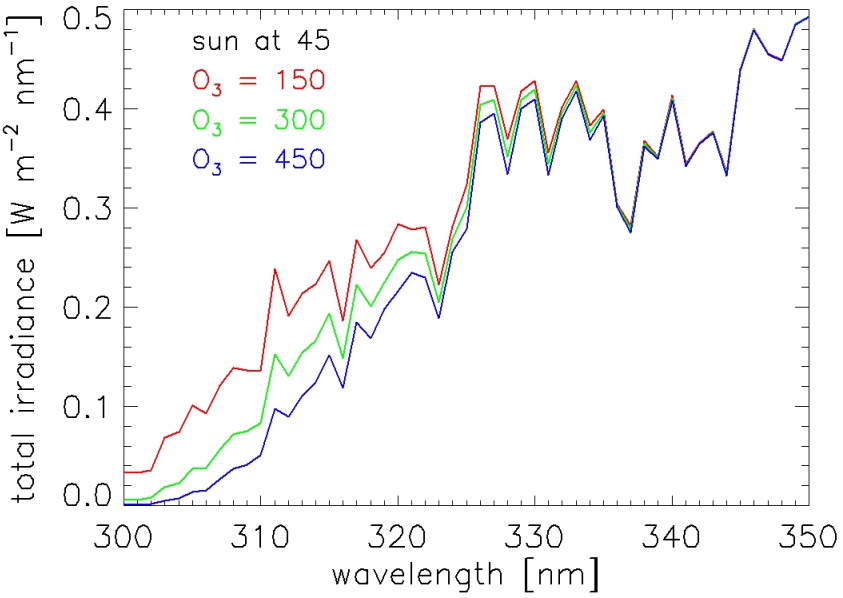

Fig. 20. Effects of ozone concentration on sea-surface total irradiances.

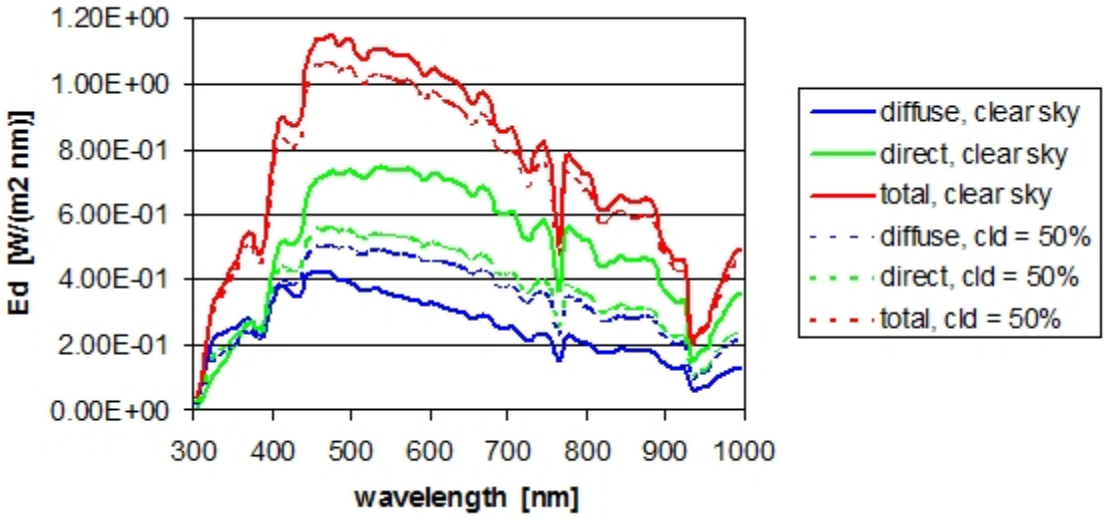

Fig. 21. Example sky irradiances for a clear sky and a 50% cloud cover.

5.2 The Empirical Sky Radiance Model

The sky radiance distribution, not just the sky irradiance, must be known in order to solve the RTE. Thus it is necessary to define the directional pattern of the sky radiance distribution. Several semianalytic models for the angular pattern of the sky radiance distribution can be found in the literature. One is probably as good as the other for most HE52 applications, because errors in the IOPs or other inputs used in a run are usually far more important than small errors in the sky radiance distribution, and because the AOPs are generally very insensitive to the pattern of the sky radiance distribution. However, some quantities, such as the surface-reflected radiance (sea surface glitter pattern) can be strongly dependent on the details of the incident radiance.

HE52 uses the Harrison and Coombes (1988) semianalytic, clear-sky model to define the relative pattern of the incident sky radiance. Figure 22 shows the angular pattern of the clear-sky radiance distribution as generated by their model for a 30 deg solar zenith angle.

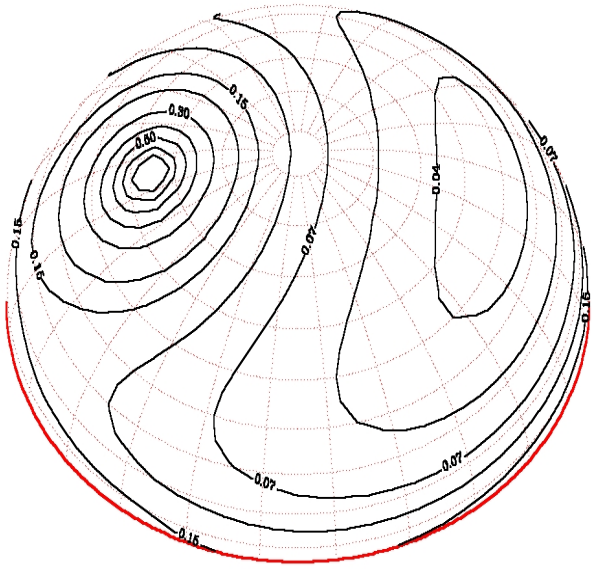

Figure 22. The Harrison and Coombes clear sky radiance angular pattern for the sun at a 30 deg zenith angle (solid lines; the sun is at the upper left of the figure). Values are relative to one near the sun. The red dotted lines show the default HydroLight quads.

HydroLight uses the direct sky irradiance in setting the magnitude of the quad-averaged sky radiance for the quad containing the sun. The relative sky radiance pattern is integrated over all quads and the diffuse sky irradiance is used to rescale this result and thereby set the magnitude of the quad-averaged diffuse sky radiances in all quads. (This is why HE52 must partition the total incident irradiance into diffuse and direct parts.) The resulting quad-averaged sun and sky radiance distribution then reproduces the direct and diffuse irradiances computed by RADTRAN-X, or as read in from a file of measured irradiances.

The irradiance and radiance routines just described, which are used when the SEMI-ANALYTICAL SKY MODEL option is selected in the UI, do a sufficiently good job of modeling the sky radiance for most applications of HE52. The use of this sky option is therefore recommended for use in general oceanographic studies

If your application depends critically on the sky radiance, then you should use a measured sky radiance distribution as input to HE52, or you should use a sophisticated atmospheric model such as MODTRAN (Acharya, et al., 1998) to generate the sky radiance input for HE52.

5.3 The Analytical Sky Radiance Model

The two routines that are selected when the IDEALIZED SKY MODEL option is selected in the UI are intended for specialized radiative transfer studies that need only a simple sky radiance distribution, e.g. a sun in a uniform background sky, or a heavily overcast sky. The sky radiance is given by

$$L_{sky}(\theta_v) = L_o\,[1 + C\cos\theta_v]\,,$$

where θ_v is the viewing direction, measured from 0 in the zenith direction. $C = 0$ gives a uniform background sky, $C = 2$ gives a cardioidal sky, and $C = 1.25$ gives a good approximation to a heavy overcast (Brunger and Hooper, 1993). Integration of $L_{sky}(\theta_v)\cos\theta_v$ gives the diffuse plane irradiance, $E_d(\text{diffuse}) = 2\pi L_o(1/2 + C/3)$, which sets the value of L_o, given $E_d(\text{diffuse}) = R_{dif}\,E_d(\text{total})$ from the user input. Here R_{dif} is the ratio of diffuse to total sky irradiance ($R_{dif} = 0$ for a sun in a black sky; $R_{dif} = 1$ for a completely diffuse sky with no sun visible). This simple sky model is intended only for use in idealized radiative transfer studies at a single wavelength. It is not available in runs at more than one wavelength.

5.4 Input of User-defined Sun and Sky Irradiances

There are three options that allow users to input their own measured or modeled sun and sky irradiances using data on Hydrolight Standard Format (HSF) data files:

1. **Input a file containing wavelength and total (sun + sky) irradiance**. HE52 will then call the built-in RADTRANX sky irradiance model to partition this total into direct (sun) and diffuse (sky) irradiances, as needed by the RTE solution code. File Sky_Irrad_Example_Edtotal.txt in the HE52\data\examples directory shows the required HSF format for such files. When using this option, the sun angle and atmospheric conditions used by RADTRANX should match as closely as possible to the conditions for the user's irradiances, so that the partition into direct and diffuse parts will be as accurate as possible.

2. **Input a file containing wavelength, direct (sun), and diffuse (sky) irradiances.** File Sky_Irrad_Example_Eddir_Eddif.txt is an example. With this option, the sun zenith angle entered in the user interface (UI) is used to determine where the sun (whose direct beam generates the user's direct irradiance values) should be placed, and the default Harrison and Coombs sky radiance model is used to determine the angular distribution of the radiance corresponding to the user's diffuse sky irradiances. However, RADTRANX will NOT be called, so the atmospheric information entered in the user interface (UI) will not be used.

2a. A special case of option 2 enables the user to **Input a lidar irradiance at a single wavelength and have the sky irradiance at all other wavelengths be set to zero.** This simulates a lidar input in an otherwise black sky. This option is discussed in more detail below. File Sky_Irrad_Example_Lidar488.txt is an example. With this option, RADTRANX will NOT be called, so the atmospheric information entered in the user interface (UI) will not be used. Keep in mind that a simulation of lidar input in HE52 corresponds to a horizontally infinite lidar footprint on the sea surface, not a small-diameter lidar beam, which gives an inherenty 3D radiative transfer problem. Also, HT5 cannot simulate time-dependent effects such as beam spreading.

3. **Input a file containing wavelength, total irradiance, and the fraction of the total that is direct irradiance.** FileSky_Irrad_Example_Edtot_frac.txt is an example. The direct irradiance will be computed from the total times the fraction, and the diffuse from the total times (1 minus the fraction). With this option, the sun zenith angle and sky radiance models are used as for option 2, and RADTRANX will not be called.

5.4.1. Running HE52 outside the 300-1000 nm range

The RADTRANX sky irradiance model is the default for computing direct and diffuse sky irradiances in HE52. The databases underlying RADTRANX cover only the 300-1000 nm range, which in turns limits HE52 to those wavelengths. However, because Option 2 does not call RADTRANX, **this option makes it possible to run HE52 over any wavelengths**. There are, however, important caveats to running HE outside of 300-1000 nm:

- Even though the user can input sun and sky irradiances for any wavelengths and run HE52, the IOPs and bottom reflectances will be given the values at the nearest wavelengths defined in their databases. For example, the concentration-specific absorption and scattering spectra found in HE52\data\defaults are defined only for 300-1000 nm, and even then the values near 300 and 1000 are often just educated guesses, because very few measurements of such things have been made outside ~350 to ~750 nm. **Thus the user must first update any needed IOP or bottom reflectance files to include values for the wavelengths of the desired run.**

- The default Harrison and Coombes model for the angular pattern of the sky radiance does not depend on wavelength, and thus becomes more and more inaccurate outside the visible wavelengths for which is was developed. Note that the sky radiance pattern is effected by Rayleigh scattering, which depends on the fourth power of the wavelength. Thus sky radiance patterns can be significantly different at near-UV wavelengths compared to near-IR wavelengths. These differences should have little effect on computed AOPs, but may become significant when computing radiances at wavelengths far from the visible.

- The index of refraction varies with wavelength, so the option to use a wavelength-dependent index should be chosen in the UI. This should have almost no effect on computed AOPs.

In any case, **if doing a run outside of 300-1000 nm, you must manually change the default wavelength limits** in file HE52\frontend\HFEdflts.txt to allow the UI to run outside its standard 300-1000 nm range. This change is made to line 3 of that file, which in the standard code reads
300,1000,1
To enable HE to run from, say, 250 to 1500 nm, change this line to
250,1500,1
and save the file. It may also be necessary to increase the default maximum number of wavelengths on the UI "change limits" form.

Figure 23 shows the total, direct, and diffuse spectral plane irradiances incident onto the sea surface as computed by the MODTRAN version 4.0 (Acharya et al., 1998) atmospheric radiative transfer code and by RADTRANX for roughly the same input atmospheric

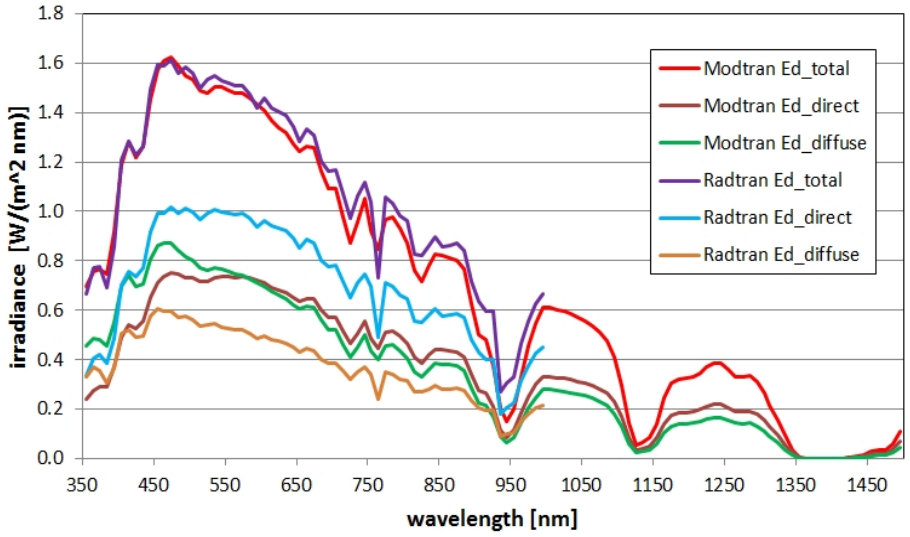

Fig. 23. Comparison of MODTRAN and RADTRANX sea-level radiances for roughly corresponding atmospheric conditions. Note that the total irradiances are close, but that the component direct and diffuse irradiances are much different for the two models.

conditions. MODTRAN was run from 350 to 1500 nm by 10 nm; RADTRANX can be run only to 1000 nm. One of the standard MODTRAN outputs is a file (`*.flx`) of computed irradiances at all levels in the atmosphere. The IDL program `Modtran_Ed_to_Hydrolight_Ed.pro` reads MODTRAN `*.flx` files, extracts the sea-level irradiances, and saves them on a HSF format file. Example file `Sky_Irrad_Example_Eddir_Eddif.txt` was created in this way. MODTRAN and RADTRANX are much different in their inputs and calculations, so it is difficult to do corresponding simulations. Nevertheless, the figure shows that the total irradiances Ed_total computed by each are within ~10% of each other except in the strong atmospheric absorption bands. However, MODTRAN and RADTRANX partition the total into direct and diffuse components in considerably different ways.

Fortunately the differences in MODTRAN and RADTRANX irradiances make almost no difference in apparent optical properties (AOPs) such as the remote-sensing reflectance R_{rs} . That is, after all, why AOPs are of value. Figure 24 shows the R_{rs} spectra for these two sky inputs and for Case 1 water with a chlorophyll concentration of 5 mg m^{-3} (using the "new Case 1" IOP model in HE52). The R_{rs} spectra are the same to within 2% at all wavelengths, and they are usually much closer.

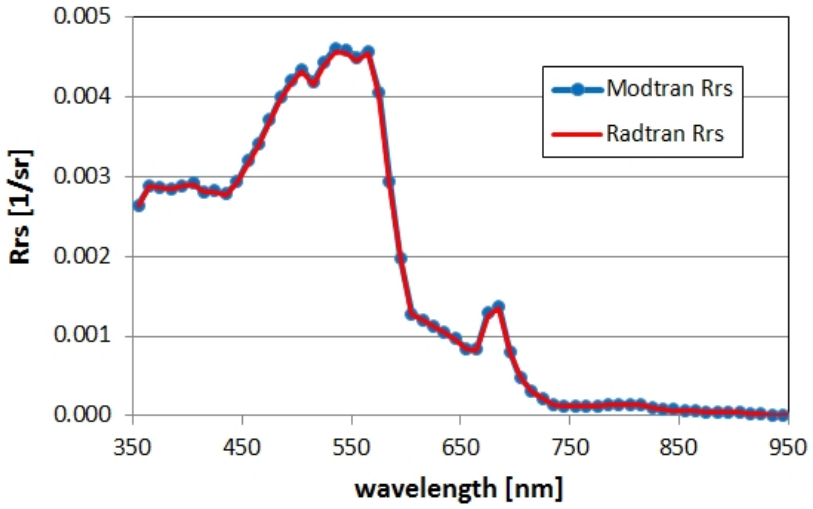

Fig. 24. Remote-sensing reflectances R_{rs} computed with the MODTRAN and RADTRANX input irradiances of Fig. 1, and Case 1 water with a chlorophyll concentration of 5 mg m^{-3}.

The code (in subroutine HE52\code\common\irradat.f) is set to read in the user's irradiances and assume that the data are values exactly at the listed wavelengths. The code then interpolates between the listed wavelengths to get sky irradiances at 1 nm resolution, and those values are then averaged over the run wavelength bands to get band-averaged irradiances (just as is done with RADTRANX values, which are at 1 nm resolution). This allows the run wavelengths to be independent of the data wavelengths. However, the values in the run printout for a given nominal wavelength will be slightly different than the values in the input data file for the same wavelength, because the printout shows the band-averaged irradiances, which depend on the size of the run wavelength bands as well as on the input data.

Because AOPs are almost unaffected by the incident sky irradiances, the main use of the new sky irradiance input options comes if radiometric variables themselves (radiances and irradiances) are to be compared with measurements or are required with great accuracy. Then it is critical to have accurate values of the input sun and sky irradiances, which set the magnitude of the entire computed light field. The other use of the new options is to allow HE52 to be run independently of RADTRANX, e.g., when it is necessary to make runs outside the standard 300-1000 nm range.

5.4.2 Simulation of Lidar-Induced Inelastic Scatter

It is also possible to use HE52 to simulate lidar-induced Raman scatter and CDOM and chlorophyll fluorescence. The code has been modified so that if the option 2 above for input of a file with Ed_direct and Ed_diffuse is selected, but there is only one wavelength in the file (not counting the negative wavelength that flags the end of data), then that input is taken to be a lidar input and all other bands in the run are given zero sky inputs, i.e., the sky is black except at the lidar wavelength.

To run a lidar simulation, do the following in the UI:

1. **Place the "sun" at the zenith** (zenith angle of 0) to simulate the lidar beam propagating straight down. (The other atmospheric parameters will not be used because RADTRANX will not be called, nor will the Harrison and Coombes sky model be called. However, the UI still requires values to be entered, since the UI doesn't know that the HE52 code won't use some inputs.)

2. Select the sky irradiance option for reading a user file containing direct and diffuse sky irradiance. **The file to be read must have the "one-wavelength" format seen in Fig. 25.** The lidar irradiance in W m^{-2} nm^{-1} should be entered as the Ed_direct value, with the Ed_diffuse value set to zero.

```
Example direct and diffuse sky spectral irradiances for LIDAR SIMULATION in a black sky.
This example assumes the run is done for a 1 nm band from 487.5 to 488.5 containing input from a 488.0 nm Lidar,
plus other bands that can be anything (e.g., 5 or 10 nm bands).
For input files having ONLY ONE input wavelength (not counting the end of file record of negative values),
the sky is automatically made black (Ed_direct = Ed_diffuse = 0.0) at all bands other than the Lidar wavelength,
which is all direct Ed (the Lidar beam and nothing else).
For this example at 488 nm, the run bands defined in the User Interface MUST include a band from 487.5 go 488.5.
Also place the sun at the zenith (zenith angle 0.0) in the User Interface.
wavelength  Ed_direct   Ed_diffuse
   (nm)     (W/m^2 nm)    (W/m^2 nm)
  488.0         1.0           0.0
   -1.0        -1.0          -1.0
```

Fig. 25. Example HSF input file for lidar simulation at 488 nm. The first 8 header records are reduced in size for printing here.

3. **On the wavelength form, it is necessary to enter a 1 nm wide band centered on the lidar wavelength, regardless of the wavelength resolution of the other bands.** You can enter all wavelength band boundaries manually, or use this trick: Enter the bands

you want for the output using the "minimim, maximum, bandwidth" option, e.g. 480 to 720 by 5 nm. Hit "continue" to go to the next form. This will enter the wavelength band list onto the wavelengths form. Then back up to the wavelengths form and select the option to enter a list of wavelengths, and manually add a 1 nm wide lidar band at the appropriate location in the list of wavelengths. (Save the inputs when you get to the final form, so that the wavelength list will be saved for subsequent runs.) The final list of wavelengths for this example should then look like

480, 485, 487.5, 488.5, 490, 495, ..., 715, 720

The red wavelengths were added manually to the list, to define the 1 nm lidar band centered at 488.0 nm. Note that the input irradiance in W m^{-2} nm^{-1} times the 1 nm bandwidth defines the lidar irradiance in W m^{-2}, as is conventional when describing lidar irradiance in a very narrow band.

This process was used in a series of EcoLight runs to simulate Raman scatter for an infinitely deep, homogeneous water body with a chlorophyll concentration of 0.05 mg m^{-3}, using the "new Case 1" IOP model. Bandwidths of 1, 5, and 10 nm were used for the nominal run bandwidths, and the RTE was solved to a depth of 50 m; the wind speed was 0. Figure 26 shows the water-leaving radiance in the region of the Raman output band centered near 582 nm. HydroLight Technical Note 10 gives a detailed discussion of this example and of interpretation of Raman-scatter output.

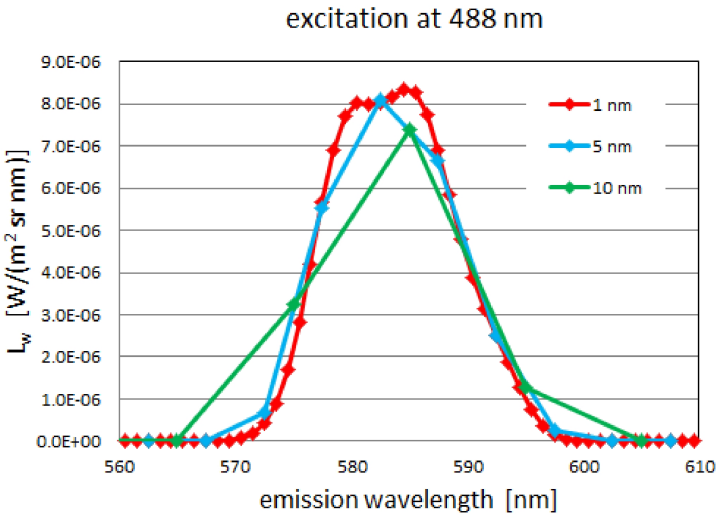

Fig. 26. Example EcoLight-computed Raman water-leaving radiances induced by a lidar input of 1 W m^{-2} at 488 nm, for three different band widths in the emission region. See HTN10 for a full discussion.

6. Inelastic Scattering Models

HE52 has the option of including three types of inelastic scattering in multi-wavelength runs: Raman scattering by water, chlorophyll fluorescence, and CDOM fluorescence. Each of these represents the transfer of light from one wavelength to another (longer) one.

In the case of Raman scattering, the way in which light is "redistributed" in wavelength is well understood. For purposes of computational efficiency, HE52 uses the azimuthally-averaged formulation of Raman scattering described in Mobley *et al*. (1993, Appendix A), which gives the correct contribution of Raman-scattered light to *irradiances*. This formulation gives some inaccuracy in the Raman contribution to an azimuthally asymmetric radiance distribution in HydroLight. The band-averaged radiances computed by EcoLight are exact. Although Raman scatter is well understood and depends only on the properties of pure water, the interpretation of the detailed spectral shape of the Raman emission as measured in the ocean is complicated by the wavelength dependence of the absorption coefficient between the emission depth and the measurement depth. See HydroLight Technical Note 10 for a discussion and examples.

In the case of chlorophyll and CDOM fluorescence, HE52 uses the computed scalar irradiance, the component absorption, and various assumptions about the fluorescence efficiency and the wavelength redistribution function to calculate the amount of light fluoresced. For chlorophyll fluorescence, the default fluorescence efficiency of 0.02 can be changed in the CHANGE DEFAULTS form of the UI. The default CDOM fluorescence quantum efficiency function is take from Hawes (1992); this function is shown in *L&W* Fig. 5.11. The revised versions (on CD) of *Light and Water* Sections 5.14 and 5.15 give the details about the calculations used for Raman scattering and fluorescence, respectively.

See §3.3 for comments regarding the depth where the bottom boundary condition is applied when the water is infinitely deep and inelastic scattering is included in the run.

Note that the option for skipping every other wavelength and filling in the missing wavelength bands by interpolation is not available with inelastic scattering is included in a run.

7. STANDARD-FORMAT DATA FILES

As seen above, HE52 comes with built-in spectra for water absorption and scattering, phytoplankton absorption, and related quantities needed for its standard IOP models. Data files with the reflectances of typical bottom materials such as sea grass, sediment, and corals are included. These default data sets are suitable for "generic" studies. There are also various *example* data files of showing how to input measured chlorophyll or mineral particle concentration profiles, IOP data (for absorption, scattering, and backscatter), and the like. However, if you wish to use HE52 to accurately model the optical environment for a particular time and location to show agreement (closure) between measured and predicted optical quantities, then the built-in data files should be replaced by data files appropriate for the chlorophyll concentration, water IOPs, bottom reflectance, etc. for your particular water body.

Each type of data required by HE52, e.g., *a* and *c* IOP data, a chlorophyll concentration profile, or a bottom reflectance spectrum, has a standard-format for that type of data. You can easily replace the HE52 default or example data with your own. You only need to put your data into the standard format for that type of data, and then browse for the name of your data file when defining the run in the HE52 user interface.

Standard-format data files are always uncompressed ASCII text files, so that they are easy to view and edit, and are independent of the computer operating system. The standard file format consists of ten header records, which can be used to identify the data file or make any other comments about the data, as the user desires. These records are read by HE52 (as character strings) and copied to the printout for documentation of the file read, but the information in these headers is not otherwise used by HE52. Records 11 and onward contain the actual data. The format of these subsequent records varies according to the type of data being read. HE52 will recognize an end-of-file when reading a data file, but it is recommended that the end of the data be explicitly flagged to prevent errors that can occur if blank records are accidentally inserted at the end of a file and are then read as data (this will usually cause run termination with an error message). The end of data is usually flagged by a negative depth or wavelength, as seen below. Since data are read one row at a time, you need to "flag" the entire last row (for each column). An easy way to do this is just to repeat (copy and paste) the last row of valid data, but then change the depth or wavelength to a negative number.

The names of the files containing various data are given to HE52 by entering the file name at the appropriate location in the UI. By default, HE52 looks in the HE52\data directory for data files (this occurs if you type in only the file name itself). You may wish to place your data files in other directories, in which case the UI lets you browse for the data file; the full path name is then automatically passed on to HE52.

The following sections show the standard formats for various types of data.

7.1 Concentration Data. Chlorophyll or mineral-particle concentrations as used by the IOP models have data records that consist of pairs of depth and concentration values. For example, a file of measured chlorophyll data might look like the following:

```
Example standard-format Chlorophyll data file
To be read by HE52 subroutine "chlzdata" on file chlzdata.f.
Ten header records (for identification of the user's data) are expected.
Record 11 begins with (free format) pairs of depth (in m)
and Chl (in mg Chl / m^3) values.
The last value is flagged by either an end of file or by negative
depth and Chl values.
Record 8:  unused in this example
Record 9:   "
depth (m)   Chl (mg/m^3)
 0.0        0.5
 5.0        0.7
10.0        1.2
15.0        2.1
20.0        1.5
25.0        0.4
 -1.        0.4
```

Note that this file has exactly 10 header records. (If you don't need 10 records to identify your file, enter "place holder" records (i.e., not blank lines), as was done here for records 8 and 9.) In this example, there are then six data records. The last record flags the end of the file with a negative depth. Note that the last record is just the previous record repeated, but with the depth changed to a negative number.

This same format is used for mineral-particle concentration data (see, for example, file HE52\data\examples\minzdata.txt) and for the absorption by CDOM at a given reference wavelength. Note that the concentration data must have units that are consistent with the concentration-specific absorption and scattering spectra.

7.2 Concentration-specific Absorption and Scattering Data. Concentration-specific absorption and scattering spectra are needed by some IOP models to convert chlorophyll or mineral concentrations to absorption and scattering coefficients. Such data files have the same format as just seen for concentration profiles, except that the depth is replaced by wavelength. For example, the mineral-particle mass-specific scattering spectrum for "red clay" (from file HE52\data\defaults\bstarmin_redclay.txt) looks like the following:

```
"RED CLAY" mass-specific mineral scattering coefficient, bmin*(wavelength) in m^2/gm,
for use by IOP (ab) routines.  These values times the concentration of mineral particles in
gm/m^3 give the scattering coefficient of the mineral particles in 1/m.
These values are computed from Fig 4-3B and Table 3-II of Y-H Ahn, "Proprietes optiques des
particules biologiques et minerales...," Univ. P. & M. Curie, Paris, France, 1990.  The Ahn
bb* values (Fig. 4-3B) are divided by bb*/b* = 0.0067 (Table 3-II) to get b*.  Data were
400-700 nm. WARNING: Extrapolated by eye and splines to 300 and 1000 nm for use in H5.0.
Spectrum may be unrealistic in the 300-400 and 700-1000 nm regions
wavelen   bmin*
  (nm)   (m^2/gm)
 300.0  0.80000
 305.0  0.80096
 310.0  0.80194
 315.0  0.80294
 320.0  0.80395
...and so on to...
 990.0  0.44298
 995.0  0.44148
1000.0  0.44000
  -1.0  -1.0000
```

Note that these mineral b^* values have units of m^{-2} gm, which is consistent with measuring the mineral concentration in units of gm m^{-3}. The product then gives a scattering coefficient in units of m^{-1}, as required by HE52. The end of data is flagged by a record with a negative wavelength.

7.3 Absorption and Beam-attenuation Data. A common use of HE52 is to compute light fields using absorption (a) and beam attenuation (c) coefficients as measured by a WETLabs ac-9 or ac-s (or similar) instrument as input via the MEASURED IOP DATA option. **Note that HE52 cannot process your raw ac9 or ac-s data. Before giving the ac-9 or similar data to HE52, you must perform all of the standard tasks of checking for bad data, applying scattering corrections, and smoothing or binning the data with depth or wavelength to remove as much noise as possible while retaining the relevant physics and biology.** Recall the comments associated with Fig. 1 of the Users' Guide and the ac-s caveat at the end of §2.6. Once the data have been cleaned up, they can be placed in a file in the HE52 standard format for ac-9 (or similar) data, which is illustrated below:

```
Example showing the standard format for ac-9 data,
as used for ac-3 data.
This file can be read by IOP routine "abacbb".
Record 11 contains the number of wavelengths and the nominal
wavelengths (free format).
records 12 and onward contain the (free-format) ac-9 data ordered as:
   depth, a(wavelength 1), ..., a(wavelength n),
          c(wavelength 1), ..., c(wavelength n)
A negative depth flags the end of file.
depth     a(456)    a(488)    a(532)    c(456)    c(488)    c(532)
3 456 488 532
0.832     0.18703   0.147457  0.128416  1.18221   1.07972   0.967512
0.965     0.199363  0.160162  0.139046  1.15831   1.05569   0.944565
1.054     0.194962  0.155575  0.135086  1.16072   1.05842   0.948916
1.188     0.183068  0.143292  0.123584  1.17185   1.07182   0.967417
1.277     0.175154  0.134851  0.114446  1.18305   1.08881   0.992936
1.366     0.168063  0.128259  0.108202  1.19956   1.10532   1.01611
.... many depth records omitted here....
14.405    0.0961292 0.066717  0.0661964 0.298839  0.253469  0.270901
14.583    0.0954552 0.0662783 0.0653839 0.295006  0.252532  0.270088
14.716    0.0947061 0.0652625 0.0642977 0.290185  0.253677  0.272243
14.894    0.0949196 0.0643945 0.0638672 0.288154  0.255801  0.274805
 -1.0     0.0949196 0.0643945 0.0638672 0.288154  0.255801  0.274805
```

(This example was extracted from file HE52\data\examples\ac3data.txt; see also HE52\examples\template\ac9data.txt for an example file with 9 wavelengths). Note that the file has exactly 10 header records for identification. Record 11 contains the number of wavelengths and the nominal wavelengths. Records 12 and onward contain the depth, a and c data, with one depth per record. The required units for a and c are inverse meters. The last record is just the last valid record duplicated, but with the depth set to -1.0. HE52 will stop reading the file when a record with a negative depth is read.

When HE52 reads a file of ac-9 data it performs a number of calculations:

- The *a* and *c* values are read for the discrete depths and wavelengths.
- Minor error checking is performed. For example, if the depths as read are not in increasing order, the depth records are re-ordered; values with duplicate depths are averaged. However, *such quality control should be done before giving the data to HE52*.
- $b = c - a$ is computed for each depth and wavelength.
- The discrete-depth, discrete-wavelength *a* and *b* values are fit with linear splines. See §7.6 for discussion of the spline fitting.

On all subsequent calls, the splines are used to define *a* and *b* at any depth and wavelength.

Note. Routine abacbb reads in *a* and *c*, and then computes *a* and *b*. If you would rather convert *a* and *c* to *a* and *b* during your own data processing, rewrite the abacbb routine slightly to read in *a* and *b*, and remove the $b = c - a$ conversion calculations.

Note. It is customary when processing ac9 data to first subtract out pure water values from the measured raw *a* and *c* values. **HE52 therefore assumes that any file of *a* and *c* data has had pure water subtracted out of the tabulated values. HE52 then automatically adds in the pure water values it computes the total absorption and scattering coefficients.**

WARNING. If HE52 needs values for *a* and *b* at depths or wavelengths outside the range of the original data, then the nearest measured values will be used (rather than terminating the run with an error message or using the splines to extrapolate, which can cause large errors). For example, if the first wavelength of your ac-9 is 412 nm, then the IOPs measured at 412 will be used for all wavelengths less than 412 nm. Thus you *can* start a HE52 run at 350 nm even if your ac-9 data starts at 412 nm, but you *cannot* reasonably expect to get accurate results at much less than 412 nm. See §7.8 for more discussion of how HE52 fills in missing data.

7.4. Backscatter Data. The format for backscatter data (measured, for example, by a HOBILabs HydroScat-6 or WETLabs bb-9 instrument) is similar to that for *a* and *c* data, except that the data records 12 and onward now contain the depth and backscatter coefficient values (rather than *a* and *c* values). See file HE52\data\examples\HydroScat6.txt for an example data file. The HE52 data processing, e.g., interpolating or extrapolating in depth and wavelength, is similar to that just described for ac-9 data.

Note. There does not seem to be any convention on removing pure water values when processing HydroScat-6, bb-9 or similar backscatter data (because the scattering contribution by pure water is usually a small contribution to the total, except in very clean water). The UI therefore requires you to indicate whether or not your file of backscatter data has had pure-water backscatter removed. HE52 will then know whether or not to subtract pure-water backscatter from your data in order to determine the particle backscatter fraction for use in defining a Fournier-Forand phase function.

7.5. Bottom Reflectance Data. As explained in §3.1, for finite-depth water a physical bottom is placed at depth z_{max}. *Light and Water* Eq. (4.81) then gives the needed bi-directional radiance reflectance of the physical bottom. The irradiance reflectance for a physical bottom is most easily communicated to HE52 by placing measured wavelength-dependent reflectances on a file in the HE52 standard format for bottom reflectances and then giving the file name to the UI. The format for reflectance data is the same as for concentration-specific data, except that the data records are now pairs of wavelength and reflectance values, rather than wavelength and concentration-specific *a* or *b* values. See any of the files in directory HE52\data\botmrefl for an example of such a data file.

For bottom reflectances, you can add your data file to the list of files in the UI pull-down menu for bottom reflectances. This is done as follows:

- Place the wavelength vs. reflectance data on a file with the HE52 standard format seen in the file HE52\examples\template\Rbottom.txt.

- Give the file a meaningful name, e.g. Rmud.txt, and place the file in the HE52\data\botmrefl directory.

- Use a text editor (such as Notepad or Wordpad) to add the name of your file to the

filelist.txt file in the HE52\data\botmrefl directory. The UI reads filelist.txt to get the names of the available files of bottom reflectances. The files are shown in the UI in the order listed in filelist.txt. You can edit this file to change the order of the files in the menu, or to add or remove files from the available list.

If you wish to use an analytical formula to compute the bottom reflectance given the wavelength, you can either use your formula to generate a data file to be read by rbottom.f, or rewrite routine HE52\code\common\rbottom.f to use your formula, rather than read a data file.

HydroLight can also use a non-Lambertian bottom in the finite-depth case. To do this, follow the directions in §8.4 and the example routine HE52\examples\templates\BRDFminn.f.

7.6 Sky Irradiance Data

There are three HydroLight Standard Formats for input of user-defined sky irradiance data, depending on which option is chosen in the UI, as discussed in Section 5.4.

The first option allows for input of the total sky irradiance data, which is partitioned into direct and diffuse parts by RADTRANX. The HSF data file has the format (partial output of file HE52\Data\Examples\Sky_Irrad_Example_Edtotal.txt):

```
Total (sun + sky) sea level irradiance obtained from MODTRAN output file
     C:\HE52\examples\IDL\MODTRAN_example.flx
This file is for use as HydroLight input.
Record 4: unused
Record 5: unused
Record 6: unused
Record 7: unused
Record 8: unused
wavelength   Ed_total
   (nm)      (W/m^2 nm)
  350.0    6.7579e-001
  360.0    7.1849e-001
  370.0    7.9674e-001
...many wavelengths omitted...
 1480.0    3.8328e-002
 1490.0    8.2752e-002
 1500.0   1.4052e-001
  -1.0  -1.0000e+000
```

The second option allows for input of the direct and diffuse sky irradiance components. The HSF data file has the format (partial output of file HE52\Data\Examples\Sky_Irrad_Example_Eddir_Eddif.txt):

```
Direct (sun) and diffuse (sky) sea level irradiances obtained from MODTRAN output
file
     C:\HE52_development\Modtran_Ed\MODTRAN_example.flx
This file is for use as HydroLight input.
Record 4: unused
Record 5: unused
Record 6: unused
Record 7: unused
Record 8: unused
wavelength   Ed_direct     Ed_diffuse
   (nm)      (W/m^2 nm)     (W/m^2 nm)
  350.0    2.2731e-001    4.4848e-001
  360.0    2.5410e-001    4.6440e-001
  370.0    2.9374e-001    5.0300e-001
...many wavelengths omitted...
 1490.0    5.0811e-002    3.1941e-002
 1500.0    8.4977e-002    5.5539e-002
   -1.0   -1.0000e+000   -1.0000e+000
```

The third option allows for input of the total sky irradiance and the direct fraction. The HSF data file has the format (partial output of file HE52\Data\Examples\Sky_Irrad_Example_Eddir_Eddif.txt):

```
Total (sun + sky) and fraction of direct sea level irradiance obtained from MODTRAN
output file
     C:\HE52\examples\IDL\MODTRAN_example.flx
This file is for use as HydroLight input.
frac_direct = Ed_direct/Ed_total
Record 5: unused
Record 6: unused
Record 7: unused
Record 8: unused
wavelength   Ed_total     frac_direct
   (nm)      (W/m^2 nm)     (nondimen)
  350.0    6.7579e-001      0.33637
  360.0    7.1849e-001      0.35365
  370.0    7.9674e-001      0.36868
...many wavelengths omitted...
 1490.0    8.2752e-002      0.61402
 1500.0    1.4052e-001      0.60475
   -1.0   -1.0000e+000     -1.00000
```

The format for input of one exact wavelength as needed for lidar simulation was shown in Section 5.4.2.

7.7 Bioluminescence Data

There is also a HydroLight Standard Format for reading in the strength of a bioluminescing layer as a function of depth and wavelength. The quantity on this file is the spectral source strength $S_o(z,\lambda)$ in W m^{-3} nm^{-1} as seen in *Light and Water* Eq. (5.107). When modeling bioluminesence in HE52, the bioluminescent source represents a horizontally homogeneous layer of bioluminescence. Such bioluminescence does occur in nature, e.g., in the "milky seas" of the Arabian Gulf. HE52 cannot model a localized source of bioluminescence, which gives an inherently 3D radiative transfer problem. HE52 always assumes that the bioluminescence is isotropically emitted. The first few rows and columns of the example data file HE52\data\examples\SobiolumData_10nm.txt are shown below:

```
EXAMPLE BIOLUMINESCENCE DATA showing the Hydrolight Standard Format.
The values are the bioluminescent source strength So in W/(m^3 nm)
Vales were selected to match the output from the example bioluminescence FUNCTION contained in s0biolum.
It approximates a bioluminescing layer with a gaussian wavelength distribution
simulating a typical concentration of dinoflagellates.
The bioluminescence source function is modeled as a gaussian in wavelength, with parameters:
     S0total =  4.00E-05 W/m^3     wave0 = 480.0 nm     sigma =  15.0 nm  (FWHM =  33.8 nm)
The bioluminescence source function is modeled as a combination of hyperbolic tangents in depth, with parameters:
     zupper =  10.0 m     zlower =  13.0 m     zscale =   2.0 m
Record 11 gives the number of wavelengths and the wavelengths:
      34       350.0       360.0       370.0       380.0       390.0       400.0       410.0       420.0       430.0
    5.00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
    5.50 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00 0.00000E+00
    6.00 5.86099E-30 1.51607E-27 2.51447E-25 2.67395E-23 1.82323E-21 7.97094E-20 2.23438E-18 4.01593E-17 4.62802E-16
    6.50 4.33072E-29 1.12023E-26 1.85795E-24 1.97580E-22 1.34719E-20 5.88976E-19 1.65100E-17 2.96739E-16 3.41967E-15
    7.00 3.19997E-28 8.27739E-26 1.37284E-23 1.45992E-21 9.95443E-20 4.35196E-18 1.21992E-16 2.19261E-15 2.52680E-14
    7.50 2.36439E-27 6.11597E-25 1.01436E-22 1.07870E-20 7.35510E-19 3.21556E-17 9.01374E-16 1.62007E-14 1.86699E-13
    8.00 1.74655E-26 4.51782E-24 7.49300E-22 7.96826E-20 5.43315E-18 2.37531E-16 6.65837E-15 1.19673E-13 1.37913E-12
    8.50 1.28778E-25 3.33110E-23 5.52479E-21 5.87521E-19 4.00600E-17 1.75137E-15 4.90939E-14 8.82380E-13 1.01687E-11
    9.00 9.36748E-25 2.42309E-22 4.01881E-20 4.27371E-18 2.91402E-16 1.27397E-14 3.57116E-13 6.41856E-12 7.39684E-11
    9.50 6.20826E-24 1.60589E-21 2.66345E-19 2.83238E-17 1.93126E-15 8.44322E-14 2.36677E-12 4.25388E-11 4.90223E-10
```

7.8 Filling in Missing Data

When working with measured data, the data will be available only at discrete depths and wavelengths. However, as HE52 solves the RTE, it must know the absorption and scattering coefficients at all depths and wavelengths relevant to the run, and it must know the bottom reflectance at any wavelength for which the RTE is to be solved. Therefore, some sort of interpolation or extrapolation must be done to define concentrations, IOPs, and bottom reflectances at depths and wavelengths other that those actually measured. HE52 does this in two ways, depending on whether the value needed falls within or outside of the range of

measurements. For example, suppose you have ac9 measurements at 9 wavelengths from 412, 440, ..., 715 nm. If HE52 needs an IOP at 420 nm, then it will interpolate between the measured values at 412 and 440. However, if it needs a value at 390, outside the range of measurements, then it will extrapolate below 412 by simply using the value at 412.

7.8.1 Interpolation within the measurement range

In the original release of HydroLight version 4.0, cubic splines were used to interpolate in single-variable data (e.g., chlorophyll as a function of depth, or reflectance as a function of wavelength), and bi-cubic splines were used to interpolate in depth and wavelength (e.g., with IOP data). However, some users came to grief with this interpolation scheme because their data were not sufficiently "smooth" for the successful use of cubic splines. (To use cubic spline fitting the data must be continuous in *at least* the first derivative.) The essence of the problem is illustrated in Fig. 27, which shows linear and cubic spline fits to hypothetical depth vs. concentration data. In the left panel of Fig. 23, the data vary relatively smoothly with depth, and there is not much difference in the linear and cubic spline fits. However, in the right panel Fig. 27, the data are noisy and there is a sharp change in concentration near a depth of 15. Now the spline fit shows large "overshoots," which do not represent the likely physical behavior of the concentration data (and in some cases, especially near sharp gradients, may make the spline-defined concentration go negative). It must be remembered that an interpolation scheme (e.g., cubic splines) is guaranteed to fit the data exactly at the measured points, but the values generated in between the measured values can be almost anything (depending on the interpolation scheme and the smoothness of the data).

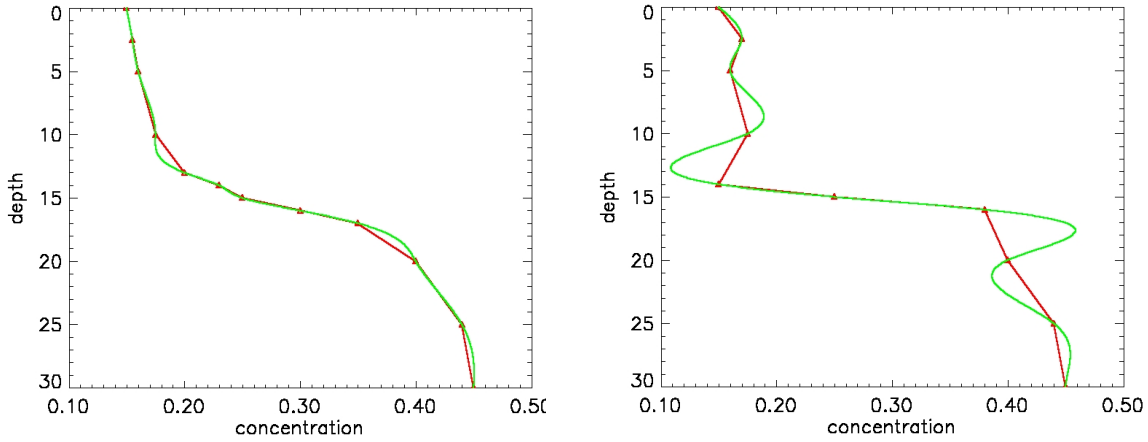

Fig. Fig. 27. Hypothetical discrete depth vs. concentration data (symbols) as fit by linear interpolation (red lines) and cubic-spline interpolation (green lines). In the left panel, the concentration varies smoothly with depth. In the right panel, there is a strong gradient in concentration near a depth of 15, and the cubic spline overshoots above and below this depth.

Because of the potential difficulties with cubic-spline interpolation, HE52 by default uses linear interpolation for *all data*. (Cubic spline interpolation is still available as an option for use by users who understand the smoothness requirements on the data to be fit. The cubic-spline code is commented out in the HE52 source code, but can be reconstituted if desired.) Linear interpolation never overshoots. However, the numerical algorithms on rare occasions can have problems "stepping past" very sharp discontinuities. This problem can be avoided by replacing sharp steps (discontinuities in the data) with slightly "rounded" steps.

Figure 28 shows an example of the absorption coefficient $a(\lambda)$ measured by an ac9 (water absorption has been removed). The green line shows the values of $a(\lambda)$ as obtained by linear interpolation between the measured values (shown by diamonds). The green curve is the values used by HE52 to solve the RTE at any wavelength between 412 and 715 nm.

7.8.2 Extrapolation outside the measurement range

However, suppose the run is done from 300 to 800 nm. There are no measured IOPs below 412 or beyond 715 nm. Therefore, HE52 will use the measurements at the nearest wavelength for the values below 412 and above 715. These values are shown in red in Fig. 28.

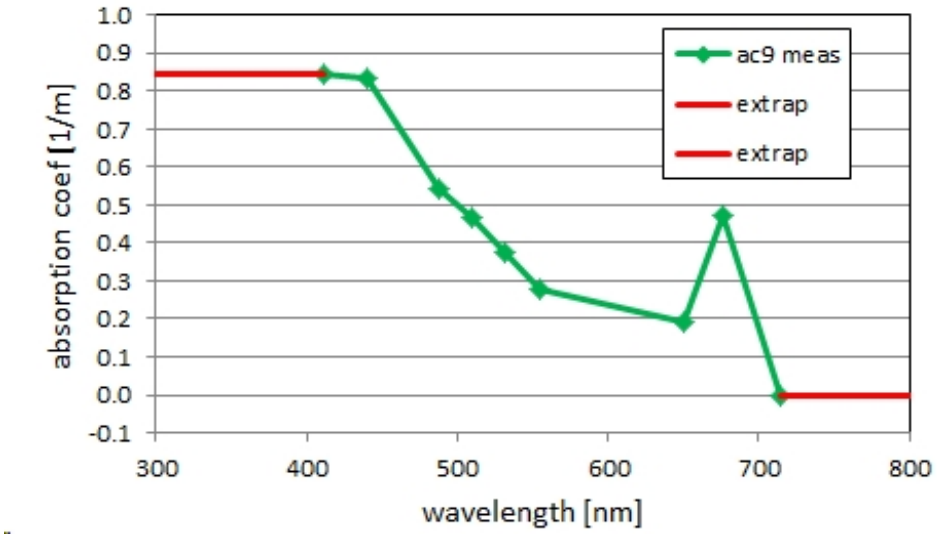

Fig. 28. Measured ac9 absorption coefficient (symbols) as fit by linear interpolation (green lines) within the range of measurement and as extrapolated outside this range (red lines).

The extrapolation below 412 in Fig. 28 will allow HE52 to run, but it is very unlikely that the results will be meaningful because the real absorption curve almost certainly does not look like the flat red curve below 412. Extrapolation much outside the range of measured data is almost guaranteed to be incorrect unless done with a physically based model to define the extrapolation in a way that is consistent with the know characteristics of the data. That is not the case in this example. About the only time such a naive extrapolation can be justified in HE52 is if a run needs to be started at 360 nm, say, in order to approximately include the effect of Raman scatter or CDOM fluorescence on the light field from 400 nm onward, and then only the solution from 400 onward will be considered as valid.

Similarly, if your IOP data start at a depth of 5 m and stop at 22 m, for example, then the values at 5 m will be used from the surface down to 5 m, and the values at 22 m will be used at all depths greater than 22 m. Again, that may or may not give good results, depending on how homogeneous the water is.

It must be kept in mind that if you want to run HE52 using measured data, then the measurements should cover the range of depths and wavelengths of interest for the simulation.

8. WRITING SUBROUTINES FOR IOP AND OTHER MODELS

As seen previously, HE52 comes with many built-in subroutines for IOP models, sky models, etc. You can use these models as they are, and most users will find these built-in models to be adequate. However, some users will want to write IOP models of their own, either to read in data on a different format or to use different bio-geo-optical models to convert measured chlorophyll or mineral profiles into the absorption and scattering coefficients required by HE52.

This section describes how to write your own subroutines for various models. The HE52 source code is written almost entirely in ANSI-standard FORTRAN 77, in order to make it easily portable to almost any computer with a FORTRAN 95 compiler (some new routines in HE52 use FORTRAN 95). All input and output files are written as ASCII text files, to assure easy transfer of files between computers with different operating systems. Writing new subroutines for an IOP model, for example, does require knowledge of FORTRAN. However, HE52 comes with templates that show you how to "fill in the blanks" to create IOP and other models, so this process is relatively easy even for inexperienced programmers. The built-in IOP models, for example, are just specific implementations of the template for "ab" models, which is file HE52\examples\templates\ab.f. However, writing an IOP model can be complicated if there are many components and different models or data files for specifying the absorption and scattering by those components. The best way to learn how to write a IOP (or other) subroutine is to compare the appropriate template file (all of which are on the HE52\examples\templates directory) with a corresponding existing subroutine (such as HE52\code\common\abnewcase1.f). Making the effort to write new routines in the same style as the templates will help you to develop routines that best mesh with existing formats and methods, which will significantly simplify any code changes and allow the new routines to be utilized within HE52 in a straightforward manner. The following sections give a few additional words of advice on writing your own routines.

User-written subroutines for IOP models, sky radiance models, etc. MUST be placed in the HE52\code\user directory. This helps keep user code separate from the distributed code. The HE52 UI will detect the presence of new routines in the user directory, or any changes to existing routines in the HE52\code\Hydrolight and HE52\code\Ecolight directories, recompile the routines as necessary, and create new executables. However, ***at least make sure your routines compile correctly before trying to run them in HE52***. This avoids confusing "HE52 bugs" with bugs in your own code. This is easily done by opening a command window, going to the directory containing the new or

modified routine, and typing the command

```
LF95 -c filename.f
```

where filename.f is the name of your new routine, or of the one you modified. If there are error messages, continue debugging your code until it compiles correctly. If the compilation is successful, then you can run HE52 and begin the process of figuring out if your new code is actually doing what you want it to do. **Section 10 gives additional advice on debugging new code and tracking down errors that cause run termination before error messages can be written.**

8.1 Subroutines for IOP Models

The most common and important task in tailoring HE52 to a specific water body is writing a subroutine to provide HE52 with the IOPs—namely the absorption (a) and scattering (b) coefficients, and the scattering phase function—of the water body. The HE52 software package comes with six IOP subroutines for computing a and b, as described in §2. Five of these routines — abconst, abcase1, abnewCase1, abcase2, and abacbb — are suitable for use in a wide variety of studies. Users can specify individual component IOPs and concentration profiles. The abcase2 routine, for instance, is suitable for many Case 1 and Case 2 waters.

If you still need to develop your own IOP model, the main thing to remember is that the total IOPs of a water body are built up as a sum of IOPs attributable to the various components of the water body. Thus, the total absorption coefficient is computed from

$$a_{\text{total}}(z,\lambda) = \sum_{i=1}^{ncomp} a_i(z,\lambda) .$$

Here $a_i(z,\lambda)$ is the absorption coefficient of the i^{th} component of the water body, which in general is a function of the depth z and wavelength λ. A similar equation is used to compute the total scattering coefficient b. The number of components in the IOP model is *ncomp*. The default array dimensioning in HE52 allows *ncomp* to as large as 4 (if you need more than 4 components, just increase the component limit on the CHANGE LIMITS form of the UI).

The process of writing your own IOP subroutine is straightforward:

- **Decide what components will be included in the model, and in what order.** For example, you might let component 1 be pure water, component 2 be CDOM, component

3 be pigmented particles, component 4 be mineral particles, and component 5 be microbubbles. The order of the components is not important in building up the absorption and scattering coefficients, but it is very important when you later specify (when running the UI) the component input, e.g., which data file contains the concentration profile for each component, or which scattering phase function is to be used for each component. If component 1 is chosen to be pure water, then you **must** select the phase function for pure water for component 1; if component 4 is mineral particles, then you **must** select a phase function for component 4 that is appropriate for mineral particles, and so on.

- **Decide how you are going to compute the absorption and scattering coefficients for each of your components.** For example, you can call the HE52 routine pureH2o to obtain the pure-water *a* and *b* values. You might want to use a simple formula to model the CDOM absorption coefficient and assume that CDOM is non-scattering. You might want to call HE52 subroutine chlzdata to read in a file of your chlorophyll data, and then use a bio-optical model to convert the chlorophyll value at depth *z* into the component *a* and *b* values at *z* and λ. You might want to read your own data file of mineral particle *a* and *b* values. Such decisions are yours to make.

- **Modify one of the existing IOP subroutines, or fill in the ab.f template, to turn your IOP model into a subroutine suitable for calling by HE52.** The documentation in HE52\templates\ab.f explains how this must be done.

- **Both HydroLight and EcoLight use the same format for IOP subroutines**, which are found in the HE52\code\common directory. Thus you only have to write one IOP subroutine, which is then useable by both HydroLight and EcoLight.

As explained in the ab.f template, you **must** give your IOP subroutine a unique name, e.g., "myIOPs" or "abmodel5" and save it as a file with the corresponding name, e.g., myIOPs.f, in the HE52\code\common directory. Your model for *a* and *b* is then ready for use by HE52.

To run HE52 with your IOP model, you select the USER IOP MODEL option when running the UI. Then enter the name of your routine (e.g., `myIOPs`), the number of components, and a descriptive name to be used to identify each component (e.g., pure water, phytoplankton, or bubbles. Note that *these names are used only by the UI* for prompting you

for the input for each component.). The UI will then lead you through the component-by-component specification of the absorption and scattering properties, the concentration, and the phase function. If you have named your IOP model "myIOPs" (in your file named myIOPs.f), for example, the UI will generate a call to your subroutine with the following form:

```
call myiops(z,wavenm,ncomp,acomp,bcomp,atotal,btotal)
```

(This call is placed in an automatically created file HE52\code\batch\root.for.)

There are several important points to note about your (or any) IOP subroutine:

- *An IOP subroutine gives HE52 information about the absorption and scattering coefficients, but not about the corresponding phase functions.* The phase function to be used with each component is not built into the subroutine for *a* and *b* because users often want to change the phase function being used to model a particular component (the new Case 1 IOP model is an exception). For example, you may wish to try several phase functions, each with a different amount of backscatter, when modeling a particle component. Such flexibility is a powerful feature of HE52.

- *The UI will still require you to provide concentration-specific a* and b* spectra, and concentration information for each component, even if they are defined differently in your IOP model.* If fluorescence is included, you will be asked by the UI to specify which model components correspond to the fluorescing material(s) included. For this reason, users are strongly encouraged to try to write their IOP models to work with the UI options whenever possible.

- *You can do whatever you wish within the IOP subroutine to compute the component a and b values.* HE52 requires only that the format of the call to the subroutine be fixed (i.e., the list of variables must be exactly as shown in the ab.f template file) and that the subroutine return the component and total *a* and *b* values.

- *The IOP subroutine must return values for any possible depth and wavelength that might be used in an HE52 run.* If you have data at discrete depths or wavelengths, you must use an interpolation or extrapolation scheme in the IOP routine to define the *a* and *b* values at all other possible depths and wavelengths. The HE52 core mathematical

routines call the IOP routine thousands of times and at very closely spaced depths (not just at the depths were output is to be saved) when solving the radiative transfer equation, so doing this in a computationally efficient manner can have a significant impact on the runtime.

- *HE52 uses the absorption and scattering coefficients at the wavelength band centers, not band averages.* Using *a* and *b* at the band centers is not generally a large source of error because *a* and *b* vary slowly and smoothly on a scale of 1 to 20 nm. If your *a* and *b* do vary rapidly with wavelength, then use smaller band widths for the run.

8.2 Subroutines for Chlorophyll Profiles

It is often convenient in the IOP models just discussed to be able to call a subroutine that returns the chlorophyll concentration *Chl* in mg Chl m^{-3}, given the depth in meters. The chlorophyll concentration then can be used as input to bio-optical models for *a* and *b*, as is done in the subroutine on file abcase1.f, for example. A chlorophyll subroutine will always be called to obtain *Chl* if chlorophyll fluorescence is included in the HE52 run.

Two ways of selecting the subroutine are provided with HE52. The first option is to use the chlorophyll subroutine on file HE52\code\common\chlzdata.f to read a file of chlorophyll data. This subroutine is automatically called if you select the USER-SUPPLIED DATA FILE option for either of the case1 IOP models. The name of the data file is specified in the UI, read in from the `Iroot.txt` file, and passed to the chlzdata routine. This routine reads this file of measured depth vs. chlorophyll data, fits a linear spline to the data, and uses the spline to compute the chlorophyll concentration at any depth. The HE52 standard format for the data files read by chlzdata.f is seen in §7.1 or on HE52\examples\templates\chlzdata.txt.

The other option is to write your own subroutine. This is done by selecting the USER-SUPPLIED SUBROUTINE option when running the UI and providing the name of your subroutine. If you do wish to write your own chlorophyll subroutine, follow the template on file HE52\examples\templates\chlzfunc.f, or modify either of the two distributed routines.

These two options should be adequate for most purposes. In either case, the core code will call the appropriate chlorophyll subroutine (chlzdata or your specified subroutine) for your problem. Note that chlorophyll profile must be specified for all depths relevant to your run.

8.3 Subroutines for CDOM Absorption

CDOM absorption can be specified as a function of depth and wavelength in the same manner as for absorption by chlorophyll. If CDOM is included in the run, CDOM absorption will be specified by the options selected in the UI. Note that the CDOM subroutine must return the CDOM absorption value for *all* depths and wavelengths relevant to the HE52 run.

8.4 Subroutines for Bottom BRDFs

For finite-depth water, the both HydroLight and EcoLight assume that the bottom is an opaque Lambertian-reflecting surface, which is fully described by specifying the irradiance reflectance of the bottom. However, HydroLight has the option of using a non-Lambertian bottom. The default Lambertian bottom for HydroLight is found on file HE52\code\Hydrolight\BRDFLamb.f (HydroLight by default calls this subroutine). If you wish to use a non-Lambertian BRDF for the bottom boundary, the procedure is as simple:

- Write a function subroutine defining the desired BRDF. An example of such a subroutine (which defines the Minnaert BRDF) is given on file HE52\examples\templates\BRDFMinn.f. The function should be named "BRDFbotm", just as is seen in files BRDFLamb.f and BRDFMinn.f.

- Move the file BRDFLamb.f from the HE52\code\Hydrolight directory to some other directory for safekeeping. (Note: to return later to using the Lambertian form you will need to reinstall the original BRDFLamb.f file from the safekeeping directory or from the HE52 installation CD) and add your file (e.g., add file BRDFMinn.f to the HE52\code\Hydrolight directory). The next time the HE52 UI runs it will detect the new subroutine, compile it, and incorporate your new BRDF into the HydroLight bottom calculations.

The only fundamental constraint on the bottom BRDF used in HydroLight is that the BRDF depend only on the *difference* of the incident and reflected azimuthal angles (i.e., the bottom must be azimuthally isotropic). Note also that *the irradiance reflectance of a non-Lambertian bottom depends on the incident lighting*; therefore the bottom reflectances available in the UI are not used if the bottom is non-Lambertian. **A non-Lambertian BRDF**

routine must define both the directional pattern of the bottom-reflected radiance and its magnitude. Doing this properly requires a thorough knowledge of BRDFs. There is a lengthy set of notes on BRDFs in the HE52\documents directory.

8.5 Subroutines for Sky Models

Although the sky models provided (and discussed in §5) are adequate for most purposes, the UI has an option for specifying a user-defined data file containing sky irradiance (obtained from direct measurements, or from a separate model such as MODTRAN). You can also specify a different routine to produce the radiance distribution in the UI, or write your own sky irradiance or radiance-pattern models by following the corresponding templates found on files HE52\examples\templates\skyirrad.txt and skyrad.txt. This is the approach to be taken if you wish to use measured or externally modeled sky radiances (e.g., from a separate run of an atmospheric radiative transfer model like MODTRAN).

8.6. Subroutines for Bioluminescence

If you check the BIOLUMINESCENCE option when running the UI, HE52 will read the specified data file or call the subroutine named in the appropriate UI text box to compute the bioluminescence source function S_o as a function of depth and wavelength. The function $S_o(z,\lambda)$ computed by this routine is discussed in *Light and Water*, Section 5.16 (see in particular Eq. 5.107). Subroutine HE52\code\common\s0biolum.f is an *example* of such a subroutine. As always, *the particular formulas in file s0biolum.f are just examples of how to compute S_o; each user must replace these formulas with ones that describe the internal source distribution of interest.*

Note also that when you include bioluminescence in HE52, you are simulating a *horizontally homogeneous bioluminescing layer* within the water body (whose details are specified in s0biolum.f ,or in your subroutine), not a point source of bioluminescence (such as a single disturbed organism).

9. SPECIAL RUNS

"Standard" HE52 runs are those that use the various defaults for quad and band partitioning, phase functions, and surface reflectance and transmittance files to solve the radiative transfer equation. The default files and standard runs are all that is needed by the vast majority of HE52 users.

A "special" HE52 run is required in only three circumstances:

- a new HydroLight quad or EcoLight band layout is needed
- a new surface wind speed or index of refraction is needed
- a new scattering phase function is to be prepared and added to the collection of available phase functions.

HydroLight Technical Note 2 in HE52\documents discusses the angular resolution and shows that little is gained by going to a finer quad resolution. Likewise, water surfaces describing wind speeds between $U = 0$ and 15 m s^{-1} and water indices of refraction between $n = 1.32$ and 1.38 are available with the distributed code. These U and n values cover the full range of oceanic conditions for which the Cox-Munk sea surface model is appropriate and for which HE52 should be run. Thus new surface runs are needed only if new quad/band layouts are also being generated. **We discourage all but the most advanced users with very special needs from altering the default quad/band partitioning.**

However, many users want to add their own scattering phase functions to the default set that comes with HE52, so special runs for discretizing phase functions are sometimes needed.

Because of the infrequent need for making special runs, control of these runs is not incorporated into the UI. Special runs can be made only by creating the needed (small) input files with a text editor, and perhaps also modifying a subroutine or writing a new one (e.g., to define a new phase function), and then submitting the run from a command window.

9.1 Defining New Quad and Band Partitions and Creating New Surface Files

The default HydroLight quad layout shown in Fig. 29a partitions the set of all directions into 10 deg polar angle (θ) by 15 deg azimuthal angle (φ). The corresponding EcoLight band layout in Fig. 29b has 10 deg polar angle bands. Each partition has polar caps with a 5 deg half angle. HE52 computes the radiances directionally averaged over these quads and bands. This angular resolution is a balance between the conflicting needs of having sufficiently high angular resolution in the radiance distribution and keeping the run times acceptably small.

Numerical studies show that a finer angular resolution does not change computed quantities such as irradiances or AOPs by more than roughly one percent, whereas accuracy starts to degrade for a coarser resolution (see HydroLight Technical Note 2). However, the run time is proportional the *square* of the number of quads or bands, so finer angular resolution comes at a high computational cost with almost no improvement in numerical accuracy of the quantities of interest to most users. **We therefore strongly recommend against altering the default quad and band resolutions, which are adequate for almost every problem of practical interest.**

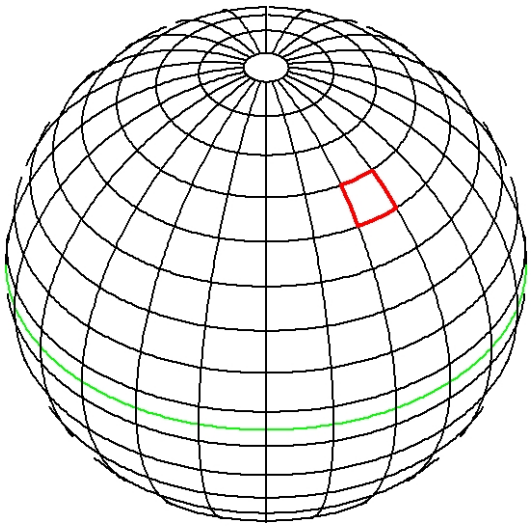

Fig. 29a. The standard HydroLight quad partitioning of all directions.

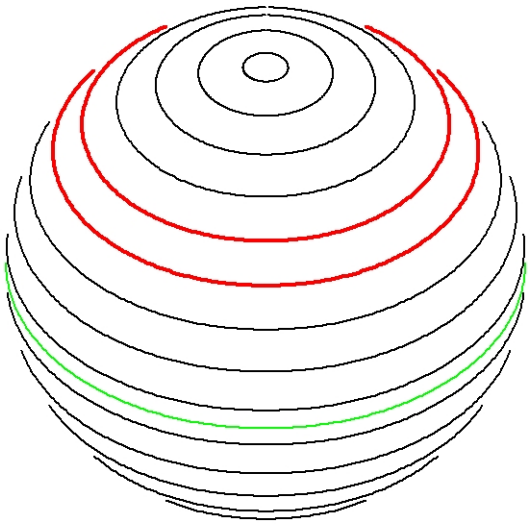

Fig. 29b. The standard EcoLight band partitioning of all directions.

Previous versions of HydroLight included the code needed for defining new quad partitions and instructions on how to create new surface files. That code is not distributed with HE52 for the simple reason that it is now a rather complex process to create new surface files for all of the wind speeds and indices of refraction, and for both the HydroLight quad and the EcoLight band partitions.

In the event that you have a problem that really does require a new quad/band partition, please contact us for user support. If we are unable to talk you out of changing the quad/band partition, then it is, quite frankly, easier for us to make the needed code modifications, do the runs, and send you the new surface files, than to explain all of the subtle code changes and how to perform the runs that loop over wind speeds and indices of refraction. Suffice it to say that a change in the quad/band partition requires re-creation of all of the surface files and rediscretization of all of the phase function files. (New surface and phase function files are needed because they depend on the quad and band partitions.) Those calculations can require a day of time to set up the runs and several days of computer time to do the calculations. **We therefore again strongly discourage users from modifying the standard HE52 quad and band partitions.**

9.2 Discretizing a Phase Function

As noted above, all phase functions must be re-discretized if the quad or band partition is changed, which is rarely if ever necessary. However, users often want to add new phase functions $\tilde{\beta}(\psi)$ to the available selection (using the standard quad or band partitions), in which case the calculations described in this section are needed. Scattering phase functions analytically defined or tabulated as pairs of scattering angle and phase function cannot be used directly by HE52. This is because the HE52 code computes radiances averaged over quads or bands. The code thus needs to compute how strongly radiance is scattered from one quad (or band, in EcoLight) to another. Exact scattering angles are thus integrated (discretized) over finite solid angles of incident and scattered directions within the quads or bands.

The phase function discretization calculations average the phase function over all pairs of quads (pairs of bands for EcoLight) as described in *Light and Water*, §8.2. The main calculation for HydroLight is the evaluation of Eq. 8.13 (and a corresponding azimuthally averaged equation for EcoLight phase functions). The HE52 code comes with many phase functions already discretized and ready for use. These phase functions are defined by subroutines in the HE52\SpecialRuns\PhaseFunction\Hydrolight and

HE52\SpecialRuns\PhaseFunction\Ecolight directories. The corresponding files of discretized phase functions are found in the HE52\data\phasefun\HydroLight and HE52\data\phasefun\EcoLight directories. The discretized phase function files have names of the form *.dpf (these files are all ASCII text files; the dpf extension just reminds us that they are files of discretized phase functions, ready for use in HE52). Note that there are separate files with the same names for HydroLight and EcoLight, in the respective directories. The HydroLight and EcoLight discretized phase function files are not interchangable because the quads and bands are different. Table 3 shows the discretized phase functions distributed with HE52.

phase function (reference)	**defining subroutine in** HE52\SpecialRuns\ PhaseFunction\...	**file containing the discretized phase function in directory** HE52\data\phasefun\...
isotropic (*L&W* Eq. 5.105)	pfiso.f	isotrop.dpf
pure water (*L&W* Eq. 3.30)	pfpure.f	pureh2o.dpf
"Petzold average particle" (*L&W* Table 3.10, column 6)	pfpart.f	avgpart.dpf
Fournier-Forand (Fournier and Forand, 1994)	pfff.f (*n* and *nu* are user-defined parameters)	FFbb0001.dpf (for 0.01% backscatter fraction) to FFbb500.dpf (for 50% backscatter fraction)
Morel large and small particle (Morel et al., 2002)	pfCase1Large.f pfCase1Small.f	pfCase1Large.dpf pfCase1Small.dpf

Table 3. Phase functions distributed with HE52.

Subroutines defining several other phase functions are available in the HE52\SpecialRuns\PhaseFunction\... directories. These include the one-term Henyey-Greenstein phase function (on subroutines name pfothg.f) and Kolelevich large and small particle phase functions (on pfLarge.f and pfSmall.f). Discretized versions of those phase functions are not provided in HE52 because they are seldom used. However, the routines provided allow these phase functions to be discretized and used in HE52, if desired.

Defining and discretizing a new phase function require several steps, which are described next. Note that all of these calculations are performed within the HE52\SpecialRuns\PhaseFunction directory and subdirectories.

- **Step 1.** Write a FORTRAN subroutine to define the desired phase function as a function of scattering angle ψ. This can be done by modifying any of the distributed routines shown in Table 3 or by following the template on file HE52\examples\templates\phasefun.f. Note that any scattering phase function $\tilde{\beta}(\psi)$ must satisfy the normalization condition

$$2\pi \int_0^{\pi} \tilde{\beta}(\psi) \sin\psi \, d\psi = 1 .$$

It is your responsibility to make sure that your phase function satisfies this normalization condition before doing the discretization calculations. Note further that the subroutine must return a value of the phase function for any scattering angle $0 \le \psi \le \pi$ (actually, for any value of $\cos\psi$, where $-1 \le \cos\psi \le 1$). After the new subroutine has been written, give the file a name of the form mynewpf.f (for example) and **save it in both the** HE52\SpecialRuns\PhaseFunction\Ecolight and the HE52\SpecialRuns\PhaseFunction\Hydrolight directory.

- **Step 2.** The routines on files HE52\SpecialRuns\PhaseFunction\EcoLight\phasef.f and HE52\SpecialRuns\PhaseFunction\HydroLight\phasef.f will call your phase function subroutine during the discretization calculations. Therefore, you must change the subroutine calls in both phasef.f files to show the name of your phase function subroutine. For example, if the current call in phasef.f is

```
call pfpart(cospsi, pfvalue)
```

and you have named your subroutine "mynewpf", you must change the above call to

```
call mynewpf(cospsi, pfvalue)
```

This change is made by opening file phasef.f with a text editor, making the change (near line 36), and re-saving the file.

- **Step 3.** Run the batch script makediscpf_EL.bat in the ..\PhaseFunction\Ecolight directory and run makediscpf_HL.bat in ..\PhaseFunction\Hydrolight. You can run these scripts by double clicking on the file names in Windows Explorer, or by entering the file name from a command window in the appropriate directory. These programs compile your new phase function subroutine and create the corresponding executable files
  HE52\SpecialRuns\PhaseFunction\discpf_EL.exe and
  HE52\SpecialRuns\PhaseFunction\discpf_HL.exe

Check to see that these two files were created. If they are not there, the compilation failed, probably because there was a bug in your phase function routine. If debugging is required, open a command window, change directories to the HE52\SpecialRuns\PhaseFunction\Hydrolight directory, and enter the command makediscpf_HL.bat. You can then see if there are any error messages during the compilation. Repeat for the EcoLight directory.

- **Step 4.** Edit the file HE52\SpecialRuns\Input\Input.txt, which has two (or more) records that are read by the discretization program. The first record gives a descriptive title, which is copied to the first record of the discretized phase function file. The second record is the root name used to generate the name of the file containing the discretized phase function. An example of these two records is

```
Measured pf from Station 4, 550 nm; standard quad partition
Sta4_550
```

If your phase function routine requires additional input, you can add additional data records after line two of the Input.txt file, and write your phase function routine to read the needed data (see file pfff.f for an example of reading a third line of input) . The root name, Sta4_550 in the above example, will be used to generate the name of the file containing the discretized phase function; this would be Sta4_550.dpf in the present example. The file extension "dpf" identifies the file as containing a discretized phase function. The dpf extension is only a convenient reminder of what is in the file; HE52 does not actually require this naming convention for phase function files. Edit the Input.txt file to change the title and file-name records as desired, and re-save the file.

- **Step 5.** Run the script HE52\SpecialRuns\PhaseFunction\discpf.bat. This can be done by double-clicking on the file name in Windows Explorer or entering the command

```
discpf.bat
```

in a command window. This script runs both of the executables discpf_EL.exe and discpf_HL.exe, which perform the calculations to create the new discretized phase function file for both EcoLight and HydroLight. The new `*.dpf` files are placed in the HE52\data\phasefun\Ecolight and HE52\data\phasefun\Hydrolight directories.

The printout from these two runs will be placed in the Printout_EL.txt and Printout_HL.txt files in the PhaseFunction directory. Inspect the printouts (with a text editor) to verify that the discretization run terminated normally. Also check the HE52\data\phasefun\Ecolight and HE52\data\phasefun\Hydrolight directories to see if the new *.dpf files are there. Finally, use a text editor to check file HE52\data\phasefun\filelist.txt to make sure that the name of your new file was added to the bottom of the list of available *.dpf files. If everything is in order, the new phase functions are now available for standard HE52 runs.

Note 1. The HydroLight discretization program adds the file name, e.g., Sta4_550.dpf, to the end of the file named filelist.txt in the HE52\data\phasefun directory. This filelist.txt file is read by the UI to generate the pulldown menu of available discretized phase functions. The files are shown in the UI in the order listed in filelist.txt. You can edit this file to change the order of the files in the menu, or to remove files from the available list.

Note 2. The discretization programs will not overwrite an existing *.dpf file. Therefore, your new phase function must have a unique name. If you want to replace an existing *.dpf file, delete it before running the discretization program.

Note 3. The discretization programs set several parameters defining the numerical accuracy of the phase function discretization. The default values are adequate for most phase functions and will not need to be changed by most users.

Note 4. The numbers in the HydroLight *.dpf files are discretized phase functions, denoted by $\tilde{\beta}(r,1,u,v)$ in *Light and Water* Chapter 8 (see Eq. 8.13), stored as a 2D array. These

numbers bear no obvious resemblance to the phase function values themselves, so don't waste your time trying to figure out what the individual numbers are. Likewise, the numbers in the files for EcoLight are band-averaged values, which are not easily interpreted.

Note 5. As previously noted in §9.1, the discretization calculations depend on the quad and band layout. If you must use a quad or band layout other than the defaults shown in Fig. 24, you must re-create all of the files of discretized phase functions (and all of the surface files). The phase functions must be rediscretized *after* creating the new surface files, because the phase function discretization routines read the quad and band layout information from the surface wind files.

Note 6. It is not currently possible to discretize wavelength-dependent phase functions and then have HE52 automatically select a different phase function at each wavelength. The only ways to have wavelength dependent phase functions are to use the power-law formula for the backscatter fraction (recall §2.3), or use the IOP data model to read in wavelength-dependent scattering and backscatter coefficients (recall §2.6). In either case, a wavelength dependent Fourier-Forand phase function will be generated.

9.3 High-theta-resolution EcoLight Runs

The standard HydroLight quad partition and EcoLight band partition of direction are shown in Fig. 29. These directional bins are adequate for almost all purposes. As shown in HydroLight Technical Note 2, outputs such as irradiances and reflectances are the same to within a percent or two if higher angular resolutions are used, but the run time increases dramatically as the number of quads or bands increases.

However, there are some simulations for which a higher angular resolution may be needed. For example, if the irradiance (or remote sensing reflectance, etc.) is being computed as a function of time over the course of a day, a plot of irradiance vs time may show a "stairstep" shape rather than a smooth curve. This is because the sun's irradiance is "smeared out" over a quad or band to computed the corresponding radiance (using the solid angle of the quad or band), even though the solar irradiance value is computed for the exact solar zenith angle. Thus when the sun passes from one quad or band to another, the smeared-out sun jumps from one theta band to another, which causes a small jump in the solar radiance used to initialize the RTE solution. These jumps are typically only a percent or less

in magnitude and will not affect AOPs, but may give an unsatisfactory appearance to some plots.

Users encountering such output have occasionally requested the ability to do higher resolution runs. EcoLight (but not HydroLight) surface files and phase functions have therefore been created as follows:

- The theta resolution is 2 deg from the equator to the pole. The polar cap has a 1 deg half angle; the band near the equator is also 1 deg.
- The same phase functions are available as in the regular version of EcoLight
- Surface files are available for the same wind speeds as the regular EcoLight (0 to 15 m s^{-1} by steps of 1 m s^{-1})
- Surface files are available only for a water index of refraction of 1.34. Therefore the surface reflectance and transmittance properties are the same for all wavelengths, water temperatures, and salinities.

The ability to do runs with the high resolution 2 deg grid is purposely not built into the HE52 UI because such runs are rarely needed. Indeed, the use of the high resolution grid is discouraged unless you absolutely do need it for some special simulation requiring highly accurate irradiances or radiances as a function of theta. Using angular resolutions higher than the standard grid is in most cases just a waste of computer time (like doing runs at 1 nm wavelength resolution unless you really do have measurements at 1 nm resolution). However, if high resolution is required, runs can be as follows.

Directory names and code must be changed from their default values. The recommended way to keep things straight is to make a copy of all of the HE52 files in a new directory called, for example, HE522deg (or reinstall HE52 from the CD, but install to a directory named HE522deg). The original standard-grid code and data will then be left unchanged and you can run either the standard or 2 deg versions, depending on which UI is run (the one in `HE52\frontend\HE52WinUI.exe`, or the one in `HE522deg\frontend\HE52WinUI.exe`.) Then make the following code changes in the new HE522deg directory:

- Rename the existing EcoLight surface directory `HE52\data\surfaces\EcoLight` to something like `HE52\data\surfaces\EcoLight_10deg`
- The EcoLight high-resolution surface files are distributed in directory `HE52\data\surfaces\EcoLight_2deg`. Rename this directory to

HE52\data\surfaces\EcoLight. This is the default directory name where EcoLight looks for surface files.

- Rename the existing EcoLight phase function directory HE52\data\phasefun\EcoLight to something like HE52\data\phasefun\EcoLight_10deg
- The EcoLight high-resolution phase function files are distributed in directory HE52\data\phasefun\EcoLight_2deg. Rename this directory to HE52\data\phasefun\EcoLight. This is the default directory name where EcoLight looks for phase function files.
- Change the mxwork array dimension parameter in routine HE52\code\EcoLight\infbotm.f from parameter(mxwork=1000) to parameter(mxwork=3200).
- The first time you run EcoLight via the UI, change the CHANGE LIMITS button on the RUN IDENTIFICATION form to increase the "Maximum number of theta quad bands..." from 20 to 92.

When EcoLight runs with these changes, the code will recompile with the new array dimensions, and EcoLight will find the 2-deg surface and phase function files when it looks under the default directory names. When you are done with the high resolution runs, be sure to reverse the above changes if you did not create a separate directory (such as HE522deg) for the 2 deg runs. That is, change HE52\data\surfaces\EcoLight, which now contains 2 deg files, back to HE52\data\surfaces\EcoLight_2deg, and change HE52\data\surfaces\EcoLight_10deg back to HE52\data\surfaces\EcoLight, and so on. Change the maximum number of theta bands back to 20 for use with the standard band partition.

Note that the 2 deg grid is available only for EcoLight runs. Correspondingly high resolution files for HydroLight are not (yet at least) available. The main reasons are (1) the very small solid angles of the HydroLight quads would require extremely large numbers of Monte Carlo photon tracings to get surface files with negligible statistical noise, especially at high wind speeds, and (2) no one has needed high resolution radiances, and the irradiances are the same for both HydroLight and EcoLight.

Finally, the 2 deg files described here have not been distributed with code versions through 5.1.4 because very few users want them, and even fewer actually need them. However, they are available on request.

10. WHEN SOMETHING GOES WRONG

HE52 standard runs are usually made by using the user interface to specify the run conditions and to automatically execute the run (including, if needed, compilation of new user-written routines). Normally, this works well: The HE52 UI opens a command window and starts the run, the code runs to completion and window then closes after execution, and the UI displays a message that the run has completed. However, when developing new code, it is almost always the case that things don't work correctly on the first try. If the new code doesn't compile or execute correctly, the run can terminate before any output (and error messages) are written to the printout (the Proot.txt file). The command window then usually closes automatically (a particularly annoying "feature" of the Windows operating system) before any error messages can be read. This can also happen if an input user data file is incorrectly formatted and causes an error in reading or processing the data, which causes the run to terminate abnormally. In these situations, it is necessary to bypass the UI and run HE52 "manually" from a command window, in which case the window remains open so that any error messages can be read. Finally, some users perform massive simulations under control of their own code for automatic generation of the needed input files, in which case the UI is not used to open the command window and run HE52.

Thus it is sometimes necessary to "manually" perform a run by explicitly issuing commands in an already open command window or by double-clicking on the appropriate file when viewed by Windows Explorer. To describe how this is done, it is assumed that

- The run input file, lroot.txt, has been created and placed in the HE52\run\batch directory

- The name of the lroot.txt file has been placed in the HE52\run\runlist.txt file.

The runlist.txt file contains the names of the one or more lroot.txt files that define the run (or runs) to be made. Multiple runs defined by different input files can be made by listing the input files in the runlist.txt file. (This is what the UI does when multiple passes are made through it via the ADD ANOTHER BATCH RUN option on the final UI form.) The lroot.txt file may have been created by the UI, by a text editor, or by the user's own program that mimics the output of the HE52 UI. There are then several ways to make a run, as follows:

Option 1. The easiest way to make a run is then to open Windows Explorer to the HE52\run directory and click on either runHL.exe to make a HydroLight run or runEL.exe to make an EcoLight run. This action will open a command window and perform the run; this is just what the UI itself does. (The executable HydroLight and EcoLight codes themselves are located in the HE52\code directory, named mainHL_stnd.exe and mainEL_stnd.exe.)

Option 2. An equivalent way to make a run is to

- Open a command window (e.g., via `start → all programs → accessories → command prompt`, in the Windows XP or 7 operating systems).

- In the command window, change directories to the HE52\run directory (via the `cd` command, e.g, `cd c:\HE52\run`) . If HE52 is installed in the C: directory, the command prompt will then be `c:\HE52\run>`

- Type runHE.exe or runEL.exe at the command prompt and hit enter. An execution window will open and the code will run just as in Option 1.

However, for both Option 1 and Option 2, the execution window will close automatically at run termination, which is undesirable if error messages appear there. The next two options allow you to make a single run and force the command window to stay open at the end of the run.

Option 3. To force the command window to stay open after run termination:

- Open a command window and cd to the HE52\run directory.

- Type the command

```
..\code\mainHL_stnd.exe < batch\Iroot.txt
```

at the command prompt and hit return (`Iroot.txt` is of course replaced by the actual name of your `Iroot.txt` file). This pipes the input file to the HydroLight executable and makes the run. The command window will remain open at the end of the run, so that any error messages can be read.

Option 4. A set of commands equivalent to Option 3 is

- Open a command window and cd to the HE52\code directory.
- Type the command

```
mainHL_stnd.exe < ..\run\batch\Iroot.txt
```

at the command prompt and hit return. This again pipes the input file to the HydroLight executable and makes the run.

Option 3 or 4 should be used when debugging new code (such as user-written IOP subroutines) or trying to determine why a run terminated abnormally, perhaps without an error message in the printout file.

Finally, it is sometimes wise to force a recompilation of all code just to make sure everything is up to date and compatible. This can happen if old and new codes (from different versions of HE52) are mixed, in which case the need for recompilation may not be recognized by the UI. To manually recompile all HydroLight code

- Open a command window and cd to the HE52\code\Hydrolight directory.

- Type the command

```
makeExeStnd_HL.bat
```

at the command prompt and hit enter. The same process is used to force recompilation of all EcoLight code via the command `makeExeStnd_EL.bat` in the HE52\code\Hydrolight directory. The actual compilation commands issued to the Lahey Fortran 95 compiler in order to compile the HydroLight code and generate mainHL_stnd.exe are found in HE52\code\Hydrolight\makeExeStnd.bat. These commands are as follows:

```
del *.obj
 copy ..\common\incfiles_default.for ..\common\incfiles_user.for
 lf95 -dbl -nco -nlst -nap -ndal -nchk -ntrace -inln -npca -nsav -stchk -o1 -nw -nwo -c
..\common\incfiles_user.for
 lf95 -dbl -chk -nco -nlst -pca -nsav -stchk -ntrace -ml winapi -win -nvsw -nw -c ..\common\w*.f90
 lf95 -dbl -chk -nco -nlst -pca -nsav -stchk -ntrace -ml winapi -win -nvsw -nw -c ..\common\*.f90
 lf95 -dbl -nco -nlst -nap -ndal -nchk -ntrace -inln -npca -nsav -stchk -o1 -nw -nwo -c *.f -ml msvb
 lf95 -dbl -co -nlst -nchk -pca -nsav -stchk -nw -nwo -c *.for -ml msvb
 lf95 *.obj -stack 750000  -ml msvb -lib ..\common\HE52info.lib -nomap -winconsole -out
mainHL_stnd.exe
 move mainHL_stnd.exe ..
del *.obj
del *.mod
```

Refer to the Lahey Fortran 95 documentation for description of these compilation options. There is a corresponding set of commands in file HE52\code\Ecolight\makeExeStnd.bat that compiles the EcoLight code.

The compilation listing on the command window goes by very quickly and can be too long for the number of lines retained for viewing in the command window. When debugging, you can save the entire listing to a text file via

```
makeExeStnd_HL.bat > complist.txt
```

You can then examine `complist.txt` with a text editor to see any error messages. Compilation error messages issued by the Lahey compiler are usually quite specific and tell you the routine, line number, and type of problem encountered.

11. REFERENCES

Journal abbreviations: AO = *Applied Optics*, JGR = *Journal of Geophysical Research*, L&O = *Limnology and Oceanography*.

Acharya, P. K., et al., 1998. MODTRAN User's Manual, Versions 3.7 and 4.0 - Draft, Air Force Research Lab, Hanscom AFB, Ma 01731-3010, 79 pages.

Ahn,Y-H, 1999. "Proprietes optiques des particules biologiques et minerales...," Ph.D. dissertation, Univ. P. & M. Curie, Paris, France.

Boss, Twardowski, and Herring, 2001. AO 40(27), 4885-4893.

Bricaud et al., 1995. JGR 100(C7), 13321-13332.

Bricaud et al., 1998. JGR 103(C13), 31033-31044.

Brunger, A. P. and F. C. Hooper, 1993. *Solar Energy*, 51(1), 53-64.

Cox, C. and W. Munk, 1954a. *J. Opt. Soc. Am.*, 44, 838-850.

Cox, C. and W. Munk, 1954b. *J. Mar. Res.*, 13, 198-227.

Fournier, G. R. and J. L. Forand, 1994. *Ocean Optics XII*, SPIE vol. 2258, 194-201.

Fournier, G. R. and M. Jonasz, 1999. Computer based underwater imaging analysis. In *Airborne and In-Water Underwater Imaging*, SPIE Vol. 3761, (corrected version).

Gathman, S. G., 1983. *Optical Engin*. 22, 57-62.

Gregg, W. W. and K. L. Carder, 1990. L&O, 35(8), 1657-1675.

Gordon, H. R., and A. Morel, 1983. *Remote Assessment of Ocean Color for Interpretation of Satellite Visible Imagery, a Review; Lecture Notes on Coastal and Estuarine Studies, Volume 4*, Springer Verlag, New York, 114 pp.

Gould, R. W., R. A. Arnone, and P. M. Martinolich, 1999. AO 38(12), 2377-2397.

Harrison, A. W. and C. A. Coombes, 1988. *Solar Energy*, 41(4), 387-392.

Hawes, S. K., 1992. Quantum fluorescence efficiencies of marine fulvic and humic acids. Master's thesis, Dept. of Marine Sci., Univ. of South Florida, St. Petersburg, FL, 92 pp.

Kasten F. and G. Czeplak, 1980. *Solar Energy*, 24, 177-189.

Kirk, J. T. O., 1994. *Light and Photosynthesis in Aquatic Ecosystems, Second Edition*, Cambridge University Press, Cambridge, 509 pages.

L&W, see Mobley (1994)

Light and Water, see Mobley (1994)

Loisel and Morel, 1998. L&O 43(5), 847-858.

Mobley, C. D., 1994. *Light and Water: Radiative Transfer in Natural Waters*, Academic Press, San Diego, 592 pp. Revised version available on CD from the author.

Mobley, C. D., 1995. *The Optical Properties of Water*, Chapter 43 in *Handbook of Optics, Second Edition, Volume I*, McGraw-Hill and Optical Society of America, 56 pages.

Mobley, C. D., B. Gentili, H. R. Gordon, Z. Jin, G. W. Kattawar, A. Morel, P. Reinersman, K. Stamnes, and R. H. Stavn, 1993. AO 32, 7484-7504.

Mobley, Sundman, and Boss, 2002. AO 41(6), 1035-1050.

Mobley, C. D., H. Zhang, and K. J. Voss, 2003. L&O 41(1), part 2, 337-345.

Mobley, C. D., D. Stramski, W. P. Bissett, and E. Boss, 2004. *Oceanography*, 17(2), 60-67.

MODTRAN, see Acharya, et al. (1998).

Morel, A., 1974. *Optical Aspects of Oceanography*. Academic Press. 1-24.

Morel, A., 1988. JGR 93(C9), 10749-10768.

Morel, A. and S. Maritorena, 2001. JGR 106(C4), 7163-7180.

Morel, Antoine, and Gentili, 2002. AO 41(30), 6289-6406.

Morrison and Nelson, 2004. L&O 49(1), 215-224.

Pope, R.M. and E.S. Fry, 1997. AO 36(33), 8710-8723.

Prieur and Sathyendranath, 1981. L&O 26(4), 671-689.

Quan and Fry, 1995. AO 34(18), 3477-3480.

Smith, R.C. and K. Baker, 1981. AO 20(2), 177-184.

Spinrad, R. W., K. L. Carder, and M. J. Perry, 1994. *Ocean Optics*, Oxford University Press, New York, 283 pages.

Twardowski et al., 2001. JGR 106, 14129-14142.

Ulloa, Sathyendranath, and Platt, 1994. AO 33(3), 7070-7077.

Vasilkov et al., 2005. AO 44(14), 2863-2869.

Voss, 1992. L&O 37, 501-509.

APPENDIX A
DETAILED DESCRIPTION OF THE RUN-TIME INPUT FOR STANDARD RUNS

The various subroutines and data files used by HE52 provide much of the information needed to make a standard run. The remaining information is read in at run time from the file lroot.txt. One such file is automatically generated by the UI for each run, and its exact format is not of interest to most users. The same lroot.txt file is used by both HydroLight and EcoLight. Increases in options and flexibility of HE52 runs has resulted in the input files becoming increasingly complicated—to the point that users are now strongly discouraged from trying to work directly with the input file.

However, some users may wish to edit these files to make small changes in the input from one run to the next (for instance, changing the wavelengths or output depths), rather than re-running the UI. This will certainly be the case for users who wish to write their own programs for generation of lroot.txt files. This is sometimes done when making many runs, or when running HE52 on operating systems othr than MicroSoft Windows, for which there is no UI. The following pages therefore describe in detail the input records found on file lroot.txt. There are 12 records or groups of records, each containing a particular type of information. Each of the records is free format. The names of the variables are those used in HE52\code\common\initial.f, which reads lroot.txt; additional documentation is given in initial.f.

RECORD 1: Default Parameters

This record contains

Name of variable: *icompile, Parmin, Parmax, PhiChl, Raman0, RamanXS, iDynZ*, RamanExp

Example: 0, 400, 700, 0.02, 488, 0.00026, 1, 5.5

icompile is the flag that tells the RUN utility whether to use the STANDARD (icompile=0, no compiling necessary) or USER (icompile=1, code must be re-compiled) executable. A root.for file will be generated and placed in code\batch **if and only if** icompile=1.

PARmin is the lowest wavelength included in PAR calculations.

PARmax is the highest wavelength included in PAR calculations.

PhiChl is the chlorophyll fluorescence efficiency.

Raman0 is the Raman reference wavelength (λ'_o in L&W Eq. 5.89).

RamanXS is the Raman scattering coefficient at the reference wavelength (a_o^R in L&W Eq. 5.89).

iDynZ is the flag that tells HE52 whether to use the Dynamic depth option if an infinitely-deep bottom boundary is selected when inelastic sources are present. If *iDynZ*=1, inelastic sources are present and an infintely-deep bottom is selected, HE52 will compute the light field to a greater depth (roughly 20 optical depths at the clearest wavelength). This helps guarantee a good solution at all depths (as described in §3.3).

RamanExp is the wavelength dependence of the Raman scattering coefficient on the excitation wavelength, as described in HydroLight Technical Note 10.

RECORD 2: Run Title

This record gives the descriptive title for the run. The title can be up to 120 characters long.

Name of variable: *ititle*
Example: Users' Guide, Example 1

RECORD 3: Rootname

This record gives the root name to be used for file generation. The name can be up to 32 characters and must not contain blanks or other "special" characters.

Name of variable: *rootname*
Example: ugex1

RECORD GROUP 4: Options

Record 4a: output options

This record specifies the output options.

Name of variables: *iOptPrnt, iOptDigital, iOptExcelS, iOptExcelM*, *iOptRad , nwskip*

Example: 0, 1, 2, 1, 1, 1

iOptPrnt is a flag for the specification of the amount of printout:

if *iOptPrnt* =-1, *Proot.txt* will contain only "minimal" runtime output.

if *iOptPrnt* = 0, *Proot.txt* will contain the "standard" output, including component IOPs and selected radiances and irradiances.

if *iOptPrnt* = 1, *Proot.txt* will contain "extensive" output including all radiance arrays and some "intermediate" values. WARNING: *this option is not recommended as it can produce a substantial amount of output*.

iOptDigital is a flag for the inclusion/omission of the digital output file:

if *iOptDigital* = 0, *Droot.txt* will not be generated for this run.

if *iOptDigital* = 1, *Droot.txt* will be generated for this run.

iOptExcelS is a flag for the inclusion/omission of the Excel Single-wavelength output file:

if *iOptExcelS* = 0, *Sroot.txt* will not be generated for this run.

if *iOptExcelS* = 2, *Sroot.txt* will be generated for this run.

iOptExcelM is a flag for the inclusion/omission of the Excel Multi-wavelength output file:

if *iOptExcelM* = 0, *Mroot.txt* will not be generated for this run.

if *iOptExcelM* = 1, *Mroot.txt* will be generated for this run.

iOptRad is a flag for the inclusion/omission of the full radiance printout file:

if *iOptRad* = 0, *Lroot.txt* will not be generated for this run.

if *iOptRad* = 1, *Lroot.txt* will be generated for this run.

nwskip sets whether to skip alternating wave bands and then interpolate to fill skipped bands (nwskip=1 solves all wavelengths; nwskip=2 solves every other wavelength, etc)

Record 4b: model options

This record specifies the models that will be used for this run and is used by `incfiles_std`.

Names of variables: *iIOPmodel, iSkyRadModel, iSkyIrradModel, iChl, iCDOM*
Example: 1, 1, 0, 2, 0

iOptmodel indicates which IOP model has been selected in the GUI:

iOptmodel = 0 for constant IOPs (IOP routine abconst.f will be called)

iOptmodel = 1 for the classic Case 1 IOP model (IOP routine abcase1.f)

iOptmodel = 2 for the Case 2 IOP model (IOP routine abcase2.f)
iOptmodel = 3 for the IOP data model (IOP routine abacbb.f)
iOptmodel = 4 for the new Case 1 IOP model (IOP routine abcase1new.f)
iOptmodel = -1 for a user-written IOP model (the user's named IOP routine in HE52\code\users will be used)

iSkyRadmodel indicates which sky radiance model has been selected in the GUI:
iSkyRadmodel = 0 for the analytical sky radiance model (single-wavelength runs only)
iSkyRadmodel = 1 for the semi-empirical sky radiance model of Harrison and Coombes (1988; routine hcnrad will be used)
iSkyRadmodel = 2 to call a user-defined sky radiance model (the user's named sky radiance routine in HE52\code\users will be used)

iSkyIrradmodel indicates how the sky irradiances are to be obtained:
If *iSkyRadmodel* = 0 then
iSkyIrradmodel = 0 flags use of the analytical sky irradiance model (single-wavelength runs only)
If *iSkyRadmodel* = 1 or 2 then
iSkyIrradmodel = 0 to call RADTRANX to obtain the direct and diffuse irradiances
iSkyIrradmodel = 1 to read a user-defined data file with the total irradiances; RADTRANX will be used to partition the total into direct and diffuse contributions (available when *iSkyRadmodel* = 1 or 2)
iSkyIrradmodel = -1 to read a user-defined data file with the direct and diffuse irradiances; RADTRANX will be NOT be called.
iSkyIrradmodel = -2 to read a user-defined data file with the total irradiances and the fraction of direct to total. RADTRANX will NOT be called

Parameters iChl and iCDOM indicate which IOP components are chlorophyll and CDOM, respectively, according to which IOP model is used. For example, when the new Case I IOP model is used, component 2 is chlorophyll ("small particles" for scattering calculations, but containing the total Chl absorption) and component 4 is CDOM. This is used when fluorescence is included in the run and absorption by chlorophyll and CDOM must be separated.

RECORD GROUP 5: IOP Specification

This group of seven records provides HE52 with information about the IOPs for the various components included in the run.

Record 5a: number of components

The first record gives the number of components expected in the IOP model (the number of components built into the "*ab*" routine) followed by the number of concentrations that will need to be read:

Names of variables: *ncomp, nconc*
Example: 2, 3

nconc ≥ *ncomp*. Normally, the number of components will equal the number of concentrations needed. The exception is when CDOM or chlorophyll fluorescence is included when CDOM or Chlorophyll are not included as model components. For example, if you include CDOM fluorescence with the Case 1 water model you will have to specify CDOM absorption in the UI and *ncomp=2* but *nconc=3*.

Record 5b: component concentrations

This record specifies the concentration of up to four components and is used only when a concentration is set to be constant with depth.

Names of variables: *compconc(j), j=1,nconc*
Example: 0, 1, 1, 0

Record 5c: Specific absorption parameters

This record consists of *nconc* lines of input where each line tells how the specific absorption will be given for each component:

Names of variables: iastropt(i), astarRef(i), astar0(i),asgamma(i)
Example: 4, 440.0, 0.06, 0.0114

The a* specifications are:

iastropt = 0: user-supplied data file read to get a* values
1: Pope & Fry absorption model used (pure water only)

2: Smith and Baker absorption model used (pure water only)
3: Prieur-Sathyenranath-Morel model used (Chlorophyll only)
4: Exponential model used (CDOM only)

The other a* parameters are only used when iastropt=4

astarRef: reference wavelength for exponential model
astar0: *a** at the reference wavelength
asgamma: exponential decay constant

Record 5d: Specific absorption data file names

The next *nconc* lines of input give the names of the files containing the specific absorption for each concentration option. The order MUST match the order of the components in the subroutine for *a* and *b*; see §8.1. Filenames that are not needed for the run may be stored with the name *astarDummy.txt* as a place holder.

Names of variable: *astarfile*(i)
Example: `astarchl.txt`
`astarcdom.txt`

Record 5e: Specific scattering parameters

This record consists of *ncomp* lines of input where each line tells how the specific absorption (line 1) and specific scattering (line 2) will be given for each component:

Names of variables: ibstropt(i), bstarRef(i), bstar0(i), coef1, coef2, coef3
Example: 1, 550, 0.3 1, 0.62, -999

The *b** specifications are:

ibstropt = 0: user-supplied data file read to get b* values
1: Power Law used (Chlorophyll only)
2: Linear (GAM) model (Chlorophyll only)
3: Gordon-Morel model used (Chlorophyll only)

If ibstropt = 1 or 2, up to five more parameters are used (if needed; otherwise, -999)

bstarRef = reference wavelength for model
bstar0 = *b** at reference wavelength
If ibstropt = 1, coef1 = *m* for power law
coef2 = *n* of the power law

If ibstropt = 2, coef1 = *m* of GAM model; the slope of the linear model, and
coef2 = *i* of GAM; the offset value in the linear model
coef3 = *n* of GAM; the exponent for the concentration

Record 5f: Specific scattering data file names

The next *ncomp* lines of input give the names of the files containing the specific scattering for each concentration option. The order MUST match the order of the components in the subroutine for *a* and *b*; see §8.1. File names that are not needed for the run (components for which *ibbopt≠0*) may be stored with the name *bstarDummy.txt* as a place holder.

Names of variable: *bstarfile*(i)
Example `bstarchl.txt`
`bstarDummy.txt`

Record 5g: type of concentrations and phase functions

This record consists of *ncomp* lines of input where each line represents the input for the i^{th} component, for *i=1* to *ncomp*

Names of variables: *itype(i), ibbopt(i), bbfrac(i), BfrefPL(i), Bf0PL(i), BfmPL(i)*,
Example: 1, 1, 0.03, 550, 0.01, 0

itype specifies what type of concentration input will be used.
itype = 0: component concentration is constant with depth
1: component concentration will be given by a user-supplied subroutine
2: component concentration will be read from a data file

ibbopt specifies how the scattering phase function will be provided.
ibbopt= 0: phase function file explicitly specified (use `pfname`)
1: construct a phase function to match a specified b_b/b ratio
±2: dynamically construct a phase function using b_b/b as specified by backscatter data (for use only with the MEASURED DATA IOP model)
+2 means that the file of backscatter data includes water value
-2 means that water values have been removed from the backscatter data
3: b_b/b specified by a power law (see §2.3), using next three input values

BfrefPL is the power law reference wavelength; used only if *ibbopt* = 3.
Bf0PL is the power law b_b/b value at the reference wavelength; used only if *ibbopt* = 3.
BfmPL is the power law *m* value; used only if *ibbopt* = 3.

Record 5h: phase function file names

The next *ncomp* number of records give the names of the files containing the discretized phase functions to be used with each component of the IOP model. The order MUST match the order of the components in the subroutine for *a* and *b*; see §8.1. Filenames that are not needed for the run (components for which *ibbopt≠0*) may be stored with the name "pfdummy.dpf" as a place holder.

Names of variables: *pfname*(1) to
pfname(*ncomp*)
Example: `pureh2o.dpf`
`avgpart.dpf`
`othgg90.dpf`

RECORD GROUP 6: Wavelengths

These records define the wavelengths to be used in the run. The first record gives *nwave*, the number of wavelength BANDS at which the model is being run:

Name of variable: *nwave*
Example: 5

The format of the next record is determined by the value of *nwave*:

if *nwave* = 0, the run is to be made at just one wavelength. In this case, the next record gives

Names of variables: *wavel, areset, breset*
Example: 532.0, 0.1, 0.25

where *wavel* is the wavelength in nm, and *areset* and *breset* are the values of *a* and *b* to be used in the abconst IOP model, if it is being used. For all other IOP models, *areset* and *breset* have values of -1.0. The sky spectral radiance at wavelength *wavel* will be used (1 nm resolution in RADTRANX).

if *nwave* ≥ 1, the run is to be made with one or more finite-width wavelength bands (> 1 nm band width, with 1 nm resolution in the sky irradiances) . In this case, the next record gives

Names of variables: *waveb*(1), *waveb*(2), ..., *waveb*(*nwave*+1)
Example: 400.0, 420.0, 430.0, 440.0, 450.0, 475.0

where the values of *waveb*(*j*) give the *nwave*+1 WAVELENGTH BAND BOUNDARIES (in nm) for which the model is to be run. (More than one line can be used if needed to list all of the band boundaries.) The *a* and *b* values as returned by the IOP model AT THE BAND CENTERS will be used. The band-averaged sky radiance will be used.

RECORD 7: Inelastic Scattering and Internal Sources

Gives the flags specifying whether or not internal sources and inelastic scatter are to be included in the run.

Names of variables: *ibiolum, ichlfl, icdomfl, iraman*, *icompchl*
Example: 0, 1, 0, 1, 2

ibiolum is a flag for the inclusion/omission of bioluminescence:
if *ibiolum* = 0, there is no bioluminescence present.
if *ibiolum* = 1, the run includes bioluminescence; user-supplied routine (s0biolum, or its replacement; see §8.6) is required.

ichlfl is a flag for the inclusion/omission of chlorophyll fluorescence:
if *ichlfl* = 0, the is no chlorophyll fluorescence present.
if *ichlfl* = 1, chlorophyll fluorescence is present; routine chlzfunc or chlzdata is called.

icdomfl is a flag for the inclusion/omission of CDOM fluorescence:
if *icdomfl* = 0, there is no CDOM fluorescence present.
if *icdomfl* = 1, CDOM fluorescence is present; routine acdom is required.

iraman is a flag for the inclusion/omission of Raman scattering:
if *iraman* = 0, there is no Raman scattering present.
if *iraman* = 1, Raman scattering is present.

icompchl is the integer index for the chlorophyll component and will equal zero if chlorophyll is not specified in the run (*icompchl* is used only when chlorophyll fluorescence is included).

RECORD GROUP 8: Sky Model

This record gives information needed by whichever sky radiance model is being used in the run. The general form of the record is

Record 8a: sky model parameters

Names of variables: *iflagsky, nsky*, *skydata*(1), *skydata*(2), ..., *skydata*(*nsky*)

iflagsky = 1 if the "idealized" sky models are being used
= 2 if the "semi-analytic" sky model is being used, with solar zenith angle being specified
= 3 if the "semi-analytic" sky model is being used, with time and location being specified

nsky is the number of values to be read in the remainder of the record.

The format of the record depends on which sky model is being used:

If *iflagsky* = 1, then *nsky* = 5 and *skydata*(1) to *skydata*(5) contain the following:

Names of variables: *suntheta, sunphi, C, rsky, Edtotal*
Example: 30.0, 0.0, 1.25, 0.3, 1.2

where

suntheta is the solar zenith angle in degrees; *suntheta* = 0.0 for the sun at the zenith and *suntheta* = 90.0 for the sun at the horizon.

sunphi is the solar azimuthal angle in degrees *relative to the wind direction*. *sunphi* = 0.0 is downwind and *sunphi* = 90.0 places the sun at a right angle to the wind. The default is *sunphi* = 0.0; see Note 2 below.

C is the cardioidal parameter [*C* in *Light and Water* Eq. (4.50)]. *C* = 0.0 gives a uniform background sky and *C* = 2.0 gives a cardioidal sky; *C* = 1.25 gives a heavily overcast sky.

rsky is the ratio of background-sky to total scalar irradiance, $0.0 \le rsky \le 1.0$. *rsky* = 0.0 for a black sky (sun only) and *rsky* = 1.0 for a fully overcast sky (no sun visible).

Edtotal is the downwelling spectral scalar irradiance due to sun and background sky incident onto the sea surface; the units are W m^{-2} nm^{-1}.

If *iflagsky* = 2, then *nsky* = 3 and *skydata*(1) to *skydata*(3) contain the following:

Names of variables: *suntheta, sunphi, cloud*
Example: 45.0, 0.0, 0.0

where

suntheta is the solar zenith angle (in degrees), defined as above.
sunphi is the solar azimuthal angle, defined as above.
cloud is the cloud cover, $0.0 \le cloud \le 1.0$. *cloud* = 0.0 for a clear sky and *cloud* = 1.0 for a solid overcast.

If *iflagsky* = 3, then *nsky* = 3 and *skydata*(1) to *skydata*(3) contain the following:

Names of variables: *hour, cloud, sunphi*
Example: 20.5, 0.5, 0.0

hour is Greenwich Mean Time in hours. Minutes are expressed as a fraction, e.g., 21.40 is 9:25 PM GMT.
cloud is the cloud cover, defined as above.
sunphi is the azimuthal angle, defined as above.

For this option, the solar zenith angle will be calculated from the specified day of year, latitude, and longitude specified in record 8b.

Note 1. If you wish to write your own sky model, you can modify the front-end programs to read in the data needed by your sky model using the *skydata* array. Then pass the needed data on to your model in the manner seen in routine initial.f.

Note 2. HE52 uses an azimuthally averaged form of the Cox-Munk wave slope equations. In this case, the azimuthal angle *sunphi* between the sun and the downwind direction is irrelevant and is set to *sunphi* = 0 by the front-end programs. If the full Cox-Munk equations are used in HE52 (which requires regenerating the surface files), the default value of *sunphi* = 0 must be changed manually in lroot.txt before running HE52.

Record 8b: atmospheric conditions

Names of variables: *jday,rlat,rlon, pres,am,rh,wv,vi,wsm,ro3*

Example: 180., 60., -20., 29.92, 5, 8, 2.5, 15, 5, -99

This record contains the atmospheric parameters that will be used to specify the earth-sun distance (*jday*), the solar zenith angle if iflagsky=3 (*jday,rlat,rlon*), ozone climatology (if *ro3*=-99), and for the Gregg & Carder atmospheric irradiance model if used (*pres,am,rh,wv,vi,wsm,ro3*).

fjday is the day of the year ("Julian" day): *fjday* = 1.0 for January 1.

rlat is the latitude in degrees; positive for north and negative for south.

rlon is the longitude in degrees; positive for east and negative for west.

pres is the average sea-level pressure (inches of Hg).

am is the marine aerosol type (1 = "marine" to 10 = "continental"; see Gathman, 1983, for a description of aerosol types.).

rh is the relative humidity (percent).

wv is the precipitable water content (cm).

vi is the average horizontal visibility (km).

wsm is the 24-hr average wind speed (m s^{-1}).

ro3 is the total ozone (Dobson units; set to -99 if a climatological value is to be used).

RECORD 9: Surface Information

This record gives the wind speed in meters per second at an anemometer height of 12 m, the index of refraction of the water, the water temperature in deg C, and the salinity in PSU.

Names of variable: *windspd, refr, temp, salinty*

Example: 5.0, 1.34, 20.0, 35.0

These values are used to select the appropriate surface reflectance and transmittance files in the data\surfaces directory. Interpolation between files is used if the windspeed is not an exact match to one of the available files. If *refr* < 0.0, then *temp* and *salinity* are used to compute the index of refraction using the formula in Quan and Fry (1995).

RECORD 10: Bottom Reflectance

This record defines the type of bottom boundary.

Names of variables: *ibotm, rflbot*
Example: 1, 0.2

ibotm is a flag for the type of bottom boundary, as follows:

if *ibotm* = 0, the water column is infinitely deep. The water below depth z_{max} (to be specified in record 10 below) is taken to be homogeneous with IOPs equal to the values at depth z_{max}. The non-Lambertian BRDF of the infinite layer of water below depth z_{max} is computed automatically from the IOPs.

if *ibotm* = 1, the bottom is an opaque Lambertian reflecting surface located at depth z_{max}. The irradiance reflectance of the bottom is taken to be *rflbot*, *independent of wavelength*. Note that $0 \le rflbot \le 1$.

if *ibotm* = 2, the bottom is an opaque Lambertian reflecting surface located at depth z_{max}. The wavelength-dependent irradiance reflectance of the bottom will be read from a HE52 standard format file of bottom reflectance data (see §7.5).

A value of *rflbot* is always read, but is used only if *ibotm* = 1.

RECORD 11: Output Depths

This record gives the depths at which output is to be saved for post-run analysis. The depths as read can be either dimensionless optical depths ζ or geometric depths z in meters. The last depth specified is taken to be the "maximum depth of interest," ζ_{max} or z_{max}, where the bottom boundary condition will be applied. (The water below the last depth is assumed to be homogeneous if *ibotm* = 0. If *ibotm* $\ge$ 1, the Lambertian bottom is placed at the last output depth.) The record has the form

Names of variables: *iop, nznom, zetanom*(1), *zetanom*(2), ..., zetanom(*nznom*)
Example: 0, 5, 0.0, 1.0, 2.0, 5.0, 10.0

iop is a flag for optical or geometric depth, as follows:

if *iop* = 0 then the *zetanom* values are GEOMETRIC depths in meters.

if *iop* = 1, then the *zetanom* values are OPTICAL depths.

nznom is the number of depths where output is desired.

zetanom(1) = 0.0, ..., *zetanom*(*nznom*) = z_{max} are the depths where output is desired.

Note that if the run contains inelastic scattering, then the run *must* use geometric depth. Note also that *zetanom*(1) must always be 0.0, the depth in the water just below the mean air-water surface, and that *zetanom*(*nznom*) is by definition the maximum depth of interest. Recall from the discussion of Section 7.1, Note 1, of the Users' Guide that a small increment will be added to each nominal output depth for the purpose of computing depth derivatives for *K* functions.

RECORD GROUP 12: Data Files

In the last group of input, the names of all of the remaining data files are specified in the following order:

1. *PureWaterDataFile*: Pure water data file (e.g., H2OabDefaults.txt)
2. *nac9Files*: The number of ac9 files to be read (either 1 or 2)
3. *ac9DataFile*: Standard-format file with unfiltered *a* and *c* data (e.g., a file of ac9 data). Used only with the IOPdata model.
4. *Ac9FilteredDataFile*: Standard-format file with *filtered a* and *c* data, which may be used when CDOM fluorescence is included
5. *HydroScatDataFile*: Standard-format data file of backscatter data (e.g., HydroScat or bb-9). Used only with IOPdata model.
6. *ChlzDataFile*: Standard-format chlorophyll profile data file (read by routine chlzdata).
7. *CDOMDataFile*: Standard-format data file containing values of CDOM absorption at a given reference wavelength (specified above)
8. *RbottomFile*: Standard-format bottom reflectance data file (read by routine rbottom)
9. *TxtDataFile*(*i*), for *i* = 1 to *ncomp*: Concentration profile data files for component *i* (*ncomp* lines of input)

10. *IrradDataFile*: Standard-format data file containing sea-surface total (sun + sky) E_d values to be used instead of RADTRAN-X values
11. *S0biolumFile*: Standard-format data file containing bioluminescent source strength (in W m^{-3} nm)

For the example of Section 6.2 of the Users' Guide, the file UGEx2.txt, which was generated by the front-end program and which contains the input for the HE52 run, is shown on the next page.

Note that dummy names were generated by the UI as place holders for the names of various files that were not needed for this run. File names for ac-9, chlorophyll, and bottom reflectance data are always read by the HE52 input routines, which do not know whether or not the files will be needed later by the core routines. In the present example, only the `Chlzdata.txt` file will be opened and read by HE52 in the course of solving the RTE.

```
0, 400, 700, 0.02, 488, 0.00026, 1, 5.3
Users Guide Example 2: Classic Case 1 IOPs
UGEx2
0,1,2,1,1,1
1,1,0,2,0
2,2
0, 1,
0, 1, 440, 0.1, 0.014
2, 3, 440, 0.1, 0.014
..\data\H2OabDefaults_SEAwater.txt
..\data\defaults\astarchl.txt
0, -999, -999, -999, -999, -999
1, 550, 0.3, 1, 0.62, -999
bstarDummy.txt
dummybstar.txt
0, 0, 550, 0.01, 0
0, 0.005, 550, 0.01, 0
pureh2o.dpf
avgpart.dpf
 35
350, 360, 370, 380, 390, 400, 410, 420, 430, 440,
450, 460, 470, 480, 490, 500, 510, 520, 530, 540,
550, 560, 570, 580, 590, 600, 610, 620, 630, 640,
650, 660, 670, 680, 690, 700,
0,1,0,1,2
2, 3, 20, 0, 0.3
-1, 0, 0, 29.92, 1, 80, 2.5, 15, 1.99809, 300
1.99809, -1.34, 20, 35
1, 0.2
0, 11, 0, 2, 4, 6, 8, 10, 12, 14, 16, 18,
20,
..\data\H2OabDefaults_SEAwater.txt
1
C:\HE52files\data\examples\ac9_data.txt
dummyFilteredAc9.txt
C:\HE52files\data\examples\Hydroscat6_withH2O.txt
C:\HE52files\data\examples\Chlzdata.txt
dummyCDOMdata.txt
dummyR.bot
dummydata.txt
C:\HE52files\data\examples\Chlzdata.txt
DummyIrrad.txt
..\data\MyBiolumData.txt
```

APPENDIX B
DESCRIPTION OF ARRAYS AND I/O UNIT NUMBERS

Within HE52, there are arrays and reserved I/O file numbers that advanced users should be aware of before modifying existing or writing new HE52 routines. Arrays *imisc* and *fmisc* are used to store miscellaneous integer and floating point variables, respectively, that are used throughout the code. The array *datafiles* stores the names of data files expected for certain components. The elements of these arrays, as well as a listing of the file unit numbers assigned for HE52 are given here.

B.1. The IMISC Array

Common block array ***imisc*** holds integer values, as follows (some values such as nphi and nL are not used by EcoLight, and some values are not used in the HE52 standard routines):

imisc index	**variable name**	**description**
1	nmu	the number of theta bands in theta = 0 to pi/2
2	nphi	the number of phi bands in phi = 0 to 2*pi
3	nL	the number of L terms in Fourier decompositions
4	nz	the number of depths where output is saved, including closely spaced pairs for K functions
5	iop	flag for optical (1) or geometric (0) depth
6	ncomp	number of components in IOP model in "ab" routine
7	nwave	number of wavelength bands
8	isource	flag for presence of internal sources or inelastic scattering (1 = present, 0 = not present)
9	idbug	flag for additional printout for debugging (most such code was removed in version 4.0)
10	nhat	dimension for spectral amplitude arrays
11	jwave	index of current wavelength band
12	ibotm	flag for bottom type (finite-depth or infinitely deep)
13	isweep	flag for sweep direction in Riccati integration
14	L	current Fourier L value
15	nuLfile	Fortran unit number for Lroot.txt printout file
16	nurad	Fortran unit number for file D_name_.txt containing the digital output for the run

17	numrays	number of initial rays started in surfwind ray-tracing code
18	nuscr1	Fortran unit number for scratch file 1
19	nuscr2	Fortran unit number for scratch file 2
20	iwrtss1	flag for writing single-wavelength spreadsheet data (files written iff = 1)
21	iwrtss2	flag for writing multi-wavelength spreadsheet data (files written iff = 1)
22	iwrtdf	Digital output file option (file written iff = 1)
23	nconc	the number of component concentrations needed (this is only relevant if CDOM or fluorescence is present)
24	iOptRad	flag for writing full radiance printout file Lroot.txt (files written iff = 1)
25	iZmax	flag to specify how zmax is determined at each wavelength
26	NwSkip	number of wavelengths to skip is skip options is selected
27	iDynZ	flags if dynamic depth solutions are allowed when sources are included and bottom is ininitely deep
28-30	<undefined>	not used in version 5.1

B.2. The FMISC Array

Common block array ***fmisc*** holds floating point values, as follows:

fmisc index	**variable name**	**description**
1	pi	3.141952654...
2	degrad	0.0174533... converts degrees to radians
3	radeg	57.2958... converts radians to degrees
4	refr	index of refraction of water (1.34 by default)
5	rad48	0.842439... = asin(1/refr) = critical angle for internal reflection by a level surface
6	relerrs	relative error for ODE integration
7	abserrs	absolute error for ODE integration
8	areset	absorption coefficient or flag for routine abconst
9	breset	scattering coefficient or flag for routine abconst
10	deltazK	the depth increment for use in generating closely spaced pairs of depths for K function computations
11	acoef(nz)	the absorption coefficient at the deepest depth
12	bcoef(nz)	the scattering coefficient at the deepest depth
13	wavel	current nominal wavelength in nm

14	rflbot	the bottom reflectance
15	wndspeed	the wind speed in m/s
16	delta	parameter in L&W Eq. 4.39
17	epsilon	parameter in L&W Eq. 4.39
18	znow	current depth in meters
19	atotal	total absorption coef the the current depth and wavelength
20	btotal	total scattering coef at the current depth and wavelength
21	a0R	Raman scattering coefficient
22	wave0	Raman scattering reference wavelength
23	PhiChl	chlorophyll fluorescence quantum efficiency
24	delbbTol	tolerance for changine phase function according to b_b/b value
25	Fo	used only by EcoLight for dynamic depth optimization
26	PARmin	min wavelength for PAR calculation
27	PARmax	max wavelength for PAR calculation
28	salinity	salinity in PSU for use in computing index of refraction
29	temp	temperature in deg C for use in computing index of refraction
30	refr	index of refraction

B.3. The DATAFILES Array

Common block array ***datafiles*** holds the filenames described in Record Group 12 of the lroot.txt file, as follows:

array index	**variable name**	**contents of file (in Hydrolight-standard format)**
0	pureH20abc	pure-water a, b, and c values to be used
1	abac9data	user-supplied UNFILTERED ac data (ac-9 or similar), for IOPdata model
2	abac9Filtered	user-supplied FILTERED ac data (ac-9 or similar), for IOPdata model
3	hscatdata	user-supplied b_b data (bb-9 or similar), for IOPdata model
4	chlzdatafile	user-supplied chlorophyll vs depth profile
5	CDOMzdata	user-supplied CDOM absorption profile
6	Rbottomdata	user-supplied bottom reflectance vs wavelength data
7 and on...	datacomp(i), for i=7 to ncomp+6	concentration data corresponding to each component

B.4. Reserved I/O Unit Numbers

Whenever possible, files are opened, read or written, closed and appended (if need be) within a single call to a subroutine. The user is encouraged to follow this example. There are a few exceptions where the unit must remain connected throughout the run. To prevent any potential conflict, the unit numbers are assigned as follows:

HE52 basic IO

3 scratch file #1
4 scratch file #2
5 Iroot (standard input)
6 Standard output (to screen, for run progress report, etc)
7 surfwind file
8 gcirrad file (sky model data file)
9 pureH2o (pure water datafile)

HE52 Output [and special project Input]

10 Proot.txt output file
11 Droot.txt output file
12 Mroot.txt output file
13 Sroot.text output file
14 [PAR model input]
15 first surfwind input file read (opened, read, and closed in `initial.f`)
16 second surfwind input file read (opened, read, and closed in `initial.f`)
17 Lroot.txt output file
18-19 reserved for future system output

User-Available unit numbers

20-39 available for use by user-written IOP models; etc.

Core HE52 datafiles (input)

40 Specific absorption data (e.g., `astarchl.txt` and `astarmin.txt`)
41 chlz datafile
42 ac9 datafile
43 bottom reflectance data file

44 not used in HE52

45 Edtotal data file (total incident irradiance as a function of wavelength)

46-49 reserved for future system data files

Discretized phase function files (*.dpf files)

50-99 There are actually ncomp files opened (units 50 to 49+ncomp) [Note: these files remain open throughout run]

APPENDIX C
LICENSE AGREEMENT FOR USE OF HYDROLIGHT-ECOLIGHT 5.2

This license agreement is also contained in the source code and in the UI.

HydroLight-Ecolight version 5.2 SINGLE-USER LICENSE AGREEMENT

HYDROLIGHT-ECOLIGHT version 5.2 is Copyright © 2013 by Curtis D. Mobley. HYDROLIGHT and ECOLIGHT are owned by Curtis D. Mobley and are protected by United States copyright laws and international treaties.

These computer programs, named HYDROLIGHT and ECOLIGHT and consisting of various main programs and subroutines, hereafter referred to collectively as "HYDROLIGHT-ECOLIGHT version 5" or as "HE52", are being licensed to the User on a non-exclusive, non-transferable license for use in scientific research.

The following requirements, which with the preceding paragraphs constitute an agreement for use of HYDROLIGHT- by the User (hereafter called "the Agreement") are to be upheld:

(1) HYDROLIGHT-ECOLIGHT version 5 may be installed on the same two computers on which the Lahey Fortran 95 Express compiler that comes with HE52 has been installed. You may not network HE52 or otherwise use it on more than two computers or computer terminals at a time.

(2) This entire notice must be retained within each main program of the source code.

(3) The following notice must be legibly displayed on the monitor or other output when HE52 is performed:

> HYDROLIGHT-ECOLIGHT version 5.2 is Copyright (c) 2013 by Curtis D. Mobley
>
> HYDROLIGHT-ECOLIGHT IS EXPERIMENTAL AND IS LICENSED "AS IS" WITHOUT REPRESENTATION OF WARRANTY OF ANY KIND, EITHER

EXPRESS OR IMPLIED. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF HYDROLIGHT-ECOLIGHT IS WITH THE USER.

(4) Title to HE52 and all portions thereof shall at all times remain with Curtis Mobley. HE52 is licensed, not sold, to the user.

(5) Any alterations, variations, modifications, additions, or improvements to HE52 or merger into other program material to form a derivative work made by the User shall be made or done at the User's own risk and expense. Any modification or combination shall not dilute or limit Curtis Mobley's rights with respect to those portions of any derivative work (in original, modified, or combined form) incorporating any portion of HE52. Any portion of HE52 included in a derivative work shall remain subject to the terms of this Agreement.

(6) The User acknowledges that HE52 is valuable to Curtis Mobley and shall be held in confidence as proprietary to Curtis Mobley, and that HE52 is licensed solely for the User's use subject to the terms of this Agreement. The User agrees not to disclose or provide HE52, in any form, to any person, party, or entity without the prior written consent of Curtis Mobley, except that the User may provide HE52 to his employees, consultants, or students as reasonably necessary to exercise his rights under this Agreement.

(7) The User agrees that HE52 (including any documentation and/or instructions for its use) is made available without warranty of any kind expressed or implied or statutory, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or conformity with whatever documentation, user manuals or other literature as may be issued by Curtis Mobley or Sequoia Scientific, Inc. from time to time. The user is warned that HE52 is not fault tolerant, and must not be relied upon in situations where financial loss, disruption of business or research, or other pecuniary loss could occur from an inability to use HE52 or from the incorrectness of its output.

(8) In no event shall Curtis Mobley and/or Sequoia Scientific, Inc. be liable for costs of procurement of substitute products or consequential damages, however caused and on any theory of liability, arising out of or related to this Agreement, even if Curtis Mobley and/or Sequoia Scientific, Inc. have been advised of the possibility of such damages.

(9) The User agrees to defend, indemnify and hold Curtis Mobley and Sequoia Scientific, Inc. harmless from any loss, cost, expense, claim, liability, demand or cause of action arising in any way from the User's use of HE52, or any products based on HE52.

(10) In the event that the User desires to develop a product, system, or service based in whole or in part on HE52 or which incorporates any portion of HE52, the User will not manufacture, sell or otherwise commercially exploit such a resultant product, system, or service before obtaining a written agreement from Curtis Mobley granting such rights, which may be granted by Curtis Mobley at his sole discretion.

(11) If you licensed HE52 in the United States, this agreement is governed by the laws of the State of Washington.

(12) This license will terminate automatically if the User fails to comply with the conditions and limitations described herein. On termination, the User must uninstall HE52 and destroy all copies of the software and documentation.

(13) It is requested, but not required, that the use of HE52 be suitably referenced or acknowledged in publications, papers, reports, presentations, or other communications for which HE52 was used as a part of the study being reported upon. It is requested, but not required, that the user of HE52 provides to Curtis Mobley a copy of all publications, reports, or other documents in which the use of HE52 is acknowledged.

THE USE OF HYDROLIGHT-ECOLIGHT VERSION 5.2 IMPLIES THE USER'S ACCEPTANCE OF THE ABOVE AGREEMENT.